

AD-A171 332

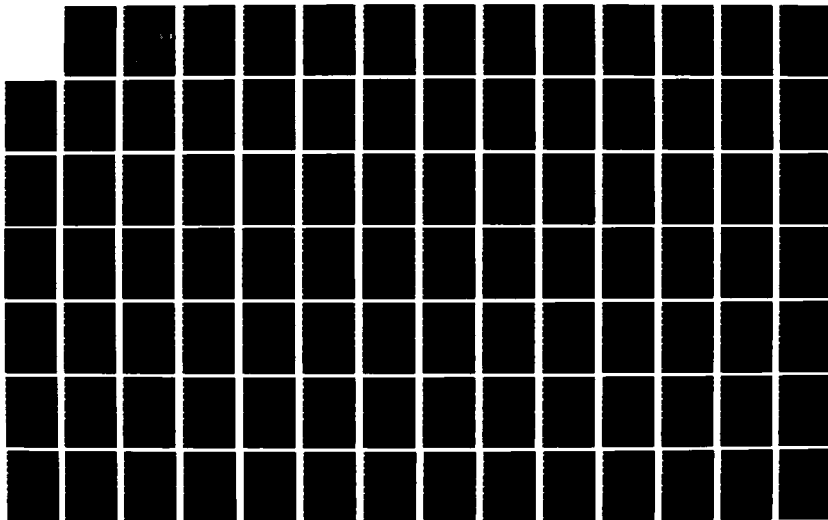
FORSYTH SECURITY MONITOR (FSM) COMPUTER PROGRAM  
DEVELOPMENT SPECIFICATION (TYPE B5)(U) LOGICON INC SAN  
DIEGO CA FEB 81 N00039-78-C-0323

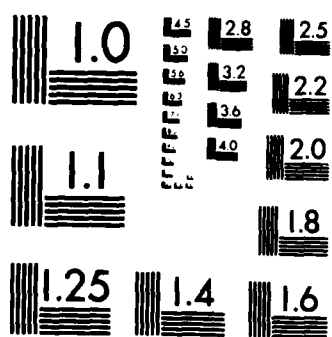
1/4

UNCLASSIFIED

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A171 332

DTK

①

ARPA-78C0323-05  
FEBRUARY 1981

**FORSCOM SECURITY MONITOR (FSM)  
COMPUTER PROGRAM DEVELOPMENT SPECIFICATION  
(TYPE B5)**

Contract Number N00039-78-C-0323

**DTIC**  
**ELECTE**  
**S D**  
AUG 26 1986  
**D**

**Prepared for:**

**Commander  
Naval Electronic Systems Command  
Washington, D.C. 20380  
Attn: Code 5703**

**Director  
Advanced Research Projects Agency  
1400 Wilson Boulevard  
Arlington, Virginia 22209**

**LOGICON**  
AUTOMATION • COMPUTERS • ELECTRONICS

APPROVED FOR PUBLIC RELEASE  
DISTRIBUTION UNLIMITED

DTIC FILE COPY

86 3 28 091

## Table of Contents

Section	Title	Page
1	SCOPE .....	1 - 1
1.1	IDENTIFICATION .....	1 - 1
1.2	FUNCTIONAL SUMMARY .....	1 - 1
1.2.1	ACCAT GUARD .....	1 - 3
1.2.2	FORSCOM Security Monitor (FSM) GUARD .....	1 - 6
2	APPLICABLE DOCUMENTS .....	2 - 1
2.1	GOVERNMENT DOCUMENTS .....	2 - 1
2.2	NON-GOVERNMENT DOCUMENTS .....	2 - 2
3	REQUIREMENTS .....	3 - 1
3.1	PROGRAM IDENTIFICATION .....	3 - 3
3.1.1	Functional Areas .....	3 - 3
3.1.2	Process/Data Structure Interfaces .....	3 - 93
3.2	DETAILED FUNCTIONAL REQUIREMENTS .....	3 - 103
3.2.1	FORSCOM Security Monitor Guardian Program (FSMGP) .....	3 - 104
3.2.2	Screener Trusted Program (SCTP) .....	3 - 119
3.2.3	FSM Assignment Daemon (FASD) .....	3 - 124
3.2.4	FSM Audit Daemon (FAD) .....	3 - 128
3.2.5	Environment Manager Program (EMP) .....	3 - 130
3.2.6	Audit Trail Display Program (ATDP) .....	3 - 136
4	QUALITY ASSURANCE .....	4 - 1
5	PREPARATION FOR DELIVERY .....	5 - 1
6	NOTES .....	6 - 1

1  
 QUALITY  
 INSPECTED

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By .....	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	



## Appendices

Appendix	Title	Page
A	FSM CONTEXT TABLES .....	A - 1
B	ENVIRONMENT MANAGER PROCESS (EMP) USER INTERFACE .....	B - 1
C	SCREENER TRUSTED PROGRAM (SCTP) USER INTERFACE .....	C - 1
D	AUDIT TRAIL DISPLAY PROGRAM (ATDP) USER INTERFACE .....	D - 1
E	GLOBAL LOGON FILE MODIFIER (USER) USER INTERFACE .....	E - 1
F	FSMGTTY PROGRAM DESCRIPTION .....	F - 1

ACCAT GUARD technology during IOC and to assess potential enhancements to this system, it is desirable to evaluate these concepts in an operational exercise.

The primary purpose of the FORSCOM GUARD experiment is to provide an IOC test bed for GUARD to test the effectiveness of GUARD as a major security application in an operational environment. The major operational exercise will be PROUD SPIRIT. Although the test will be conducted in a benign system-high environment, the experiment will be structured to simulate the "potentially threatening" environment of allowing the availability to non-Top Secret WES sites.

Presently, the MLS ACCAT GUARD system has completed its GENSER phase testing. Based on the GENSER test phase the following new GUARD requirements have been identified:

- a. An "automatic" downgrade mechanism for reducing the workload of the Security Watch Officer (SWO).
- b. A special "low-to-high" filtering mechanism for limiting low user capabilities on the system.

The first new requirement allows the GUARD, in a trusted manner, to perform automatic downgrade operations for certain recognizable high side outputs. Presently with ACCAT GUARD all high-to-low data must be screened by the SWO. This new feature will reduce the workload of the SWO and also improve the overall responsiveness of the system.

## SECTION 1

### SCOPE

#### 1.1 IDENTIFICATION

— This document sets forth design and implementation requirements for the Initial Operational Capability (IOC) phase of the GUARD project. The IOC phase of GUARD will operate under the UNIX Operating System, which shall ultimately consist of the Kernelized Secure Operating System (KSOS) security kernel and the UNIX emulation package. This IOC phase of GUARD software will operate under a version of Western Electric UNIX (Baseline 6.0) enhanced to provide both a more KSOS-like environment and operate with the special IOC testbed interfaces.

#### 1.2 FUNCTIONAL SUMMARY

The FORSCOM WWMCCS environment has been chosen for the GUARD IOC test site. This environment is ideal for experimenting further with the GUARD technology. It is desired to experiment with the GUARD technology at FORSCOM in order to determine whether FORSCOM could connect its present and future WES sites without jeopardizing security thus eliminating the efforts and associated costs of upgrading these sites to Top Secret, the system high operation. Therefore, to determine the true value of the

The second requirement is for security considerations. The purpose of the filtering mechanism is to confine the low user's activity on the high side of GUARD. It is believed that with the incorporation of the automatic downgrade mechanism, it will be necessary to restrict the low user from performing certain operations. This is particularly true for GUARD's IOC phase at FORSCOM, where the user is provided an interactive environment.

The following outcome of the GUARD IOC experiment will provide excellent data as to the true effectiveness of a downgrade mechanism for solving MLS problems in essentially non-MLS systems.

#### 1.2.1 ACCAT GUARD

The ACCAT GUARD system allows the controlled "writing down" of information from a high security level network computer host to that of a lower security level network computer host. The "writing down" operation of the system is controlled completely by a Security Watch Officer (SWO). Since the ACCAT GUARD will become one of the first applications to be placed under the Kernelized Secure Operating System (KSOS) (currently under development), it will be a verifiably secure system. In particular, the "writing down" mechanism has been formally specified and will be implemented using the Modula programming language, a language which lends itself to formal code proofs.

Functionally, the ACCAT GUARD (see Figure 1-1) provides a transactional-like interface to its user set. ACCAT GUARD users reside on distant hosts connected to the ARPANET. One user set resides on a host with a high security level and the other user set resides on a host with a lower security level. The users communicate to each other and other hosts through GUARD using standard ARPANET messages (referred to in GUARD as transactions). Three basic transactions are supported by the ACCAT GUARD.

- a. Messages from a low/high user to a high/low user.
- b. Canonical queries from a low/high user to a high/low host.
- c. English queries from a low/high user to a high/low host.

Messages are standard ARPANET mail, canonical queries are queries which are in the proper syntactic form for direct processing by the recipient host, and English queries are queries which must be manually translated by GUARD personnel into canonical queries prior to submission to the recipient host.

Communication security is guaranteed by the COMSEC approved Private-Line-Interfaces (PLI) connecting the hosts to the ARPANET. The PLI's provide encryption/decryption of information flowing from/to the hosts. The connectivity of a low host to a high host is only possible through the ACCAT GUARD. GUARD has two PLI connections to the ARPANET, one compatible with the high host PLI and the other compatible with the low host PLI (Figure 1-1).

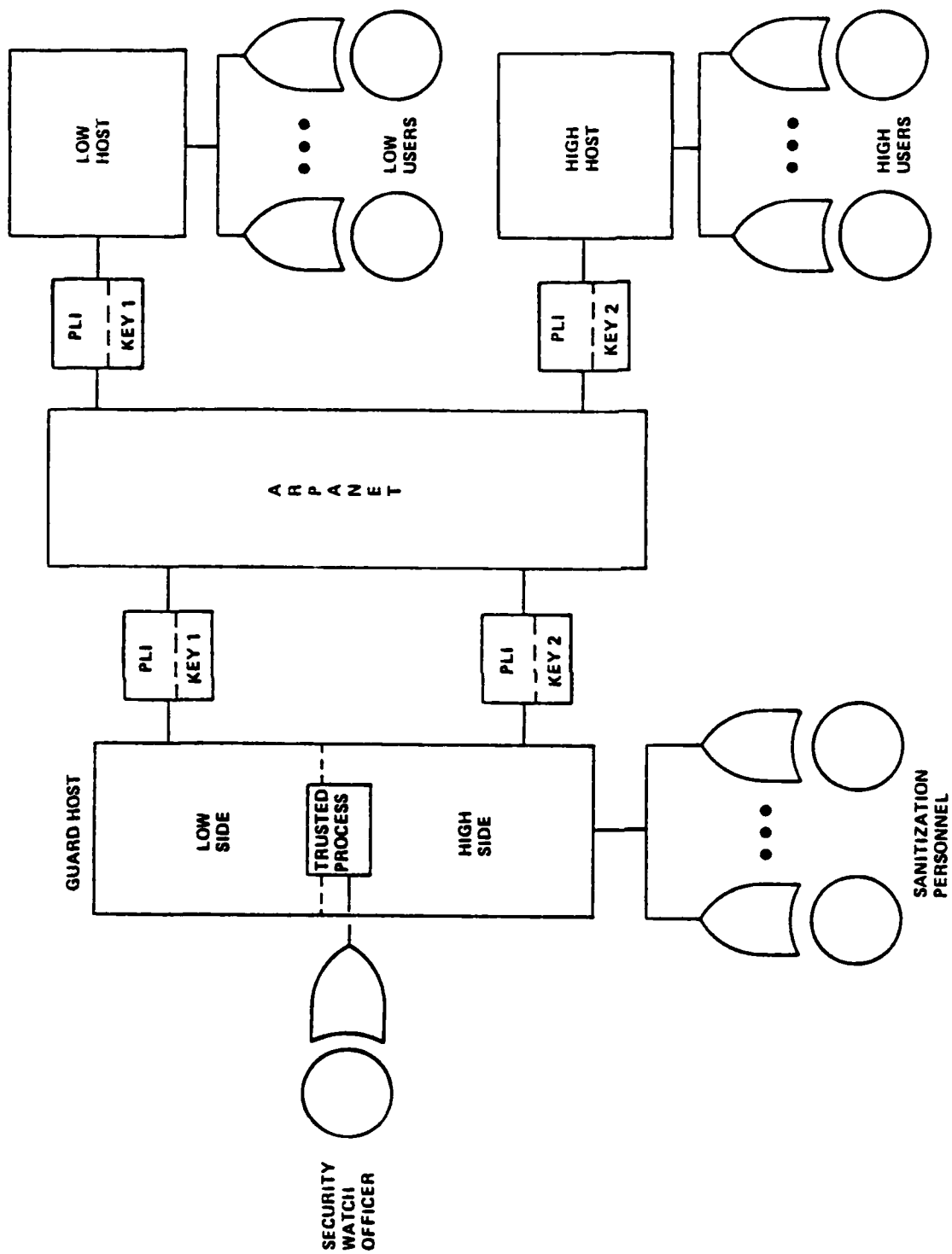


Figure 1-1. MLS ACCAT GUARD Configuration

### 1.2.2 FORSCOM Security Monitor (FSM) GUARD

The FORSCOM Security Monitor (FSM) GUARD (to be called FSM from here on) will be a further extension of the technology of the existing ACCAT GUARD and be based on the actual software currently comprising this system. The FSM system, however, requires some alterations to the current ACCAT GUARD system. These are as follows:

- a. The FSM system will interface a set of low security level users to a high security level host (see Figure 1-2).
- b. The "communication unit" of the FSM is not transaction oriented, but instead is highly interactive.
- c. Based on GENSER phase testing, the FSM requires that certain low level user inputs be filtered prior to submission to the high level host.
- d. Based on GENSER phase testing, the FSM will employ a special "recognition mechanism" which will allow the automatic screening for certain system level prompts to the low security level user.

Since the FSM exercise will be operating in a benign environment, the standard UNIX version of the ACCAT GUARD (GENSER version) can be used as a software baseline for this new system.

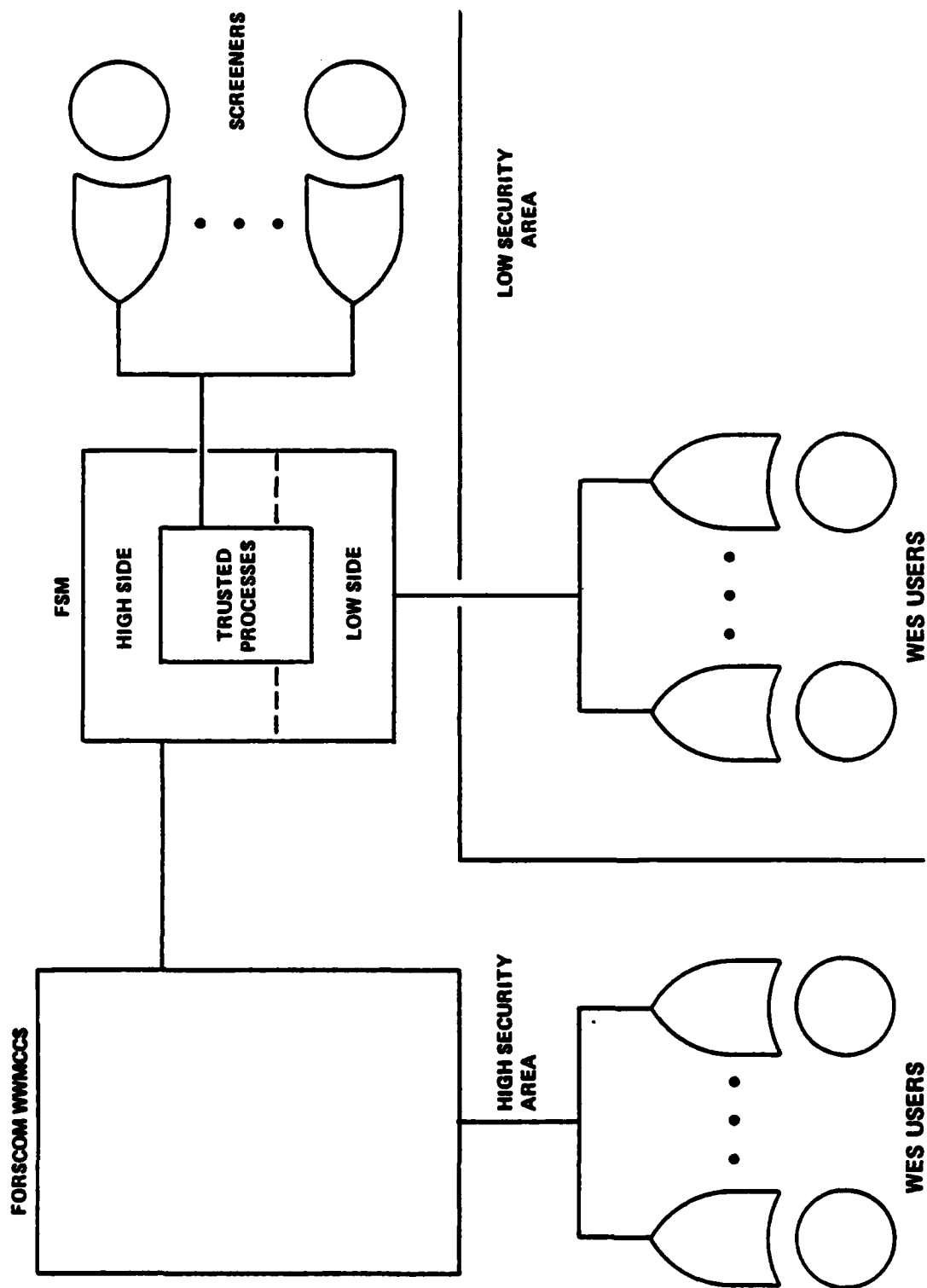


Figure 1-2. FORSCOM Security Monitor (FSM) Configuration



Based on GENSER phase testing, it has become advisable that a special automatic screening mechanism be added to the current GUARD system. Although the initial test of this technology will only simulate the "writing down" of information from a high security level to a lower security level, the basic architecture and design of this mechanism must be carefully analyzed to ensure that future implementations under KSOS can be verified to operate correctly.

The basic operation of the FSM system is as follows. As depicted in Figure 1-2, the FORSCOM WWMCCS system, the FSM and the Screener are located in a Top Secret system high area. (The Screener is the FSM counterpart to the ACCAT GUARD's Security Watch Officer.) The FORSCOM WWMCCS system and the FSM are directly connected and the low security users are connected to the FSM only. Therefore, the multi-level secure situation exists only in the FSM processor. When a low security level user logs onto the FORSCOM system, they are in reality logging onto the FSM system which in turn logs the user onto the FORSCOM system. Note, the FORSCOM system software is not cognizant of the fact that the FSM system is between it and the low security users. Therefore, the FSM system must have complete knowledge of what the user is entering into the system. This is required for two reasons:

- a. In order for FSM to properly filter what the user is entering, the filtering must be accomplished within the context of what the user is doing in the FORSCOM system.

- b. In order for FSM to properly recognize certain system responses for automatic screening sequences, FSM must again do this within the context of what the user is doing in the FORSCOM system.

A pre-established set of FORSCOM system level prompts will be recognized by the FSM system and sent directly to the low security level user without Screener intervention. All variable data (e.g., teleconferencing messages) must go through the standard screening mechanism.

## SECTION 2

### APPLICABLE DOCUMENTS

Following is a list of documents relevant to this computer program development specification.

#### 2.1 GOVERNMENT DOCUMENTS

- a. Logicon, "ACCAT GUARD Computer Program Development Specification (Type B5)", February 1979.
- b. Logicon, "Formal Specification of GUARD Trusted Software", September 1979.
- c. Logicon, "ACCAT GUARD Acceptance Test Plan (ATP)", March 1980.
- d. Mitre Corporation, "ACCAT GUARD System Specification (Type A)", August 1978.
- e. Mitre Corporation, "ACCAT GUARD Overview", November 1979.
- f. U.S. Army Forces Command, "WWMCCS Entry System (WES) Terminal User's Handbook", September 1978.

- g. IBM, "FORSCOM Computer Security Concept Experiment Notebook", April 1980.

## 2.2 NON-GOVERNMENT DOCUMENTS

- a. Bell Telephone Laboratories, "UNIX Programmer's Manual", May 1973.

## SECTION 3

### REQUIREMENTS

The FORSCOM Security Monitor (FSM) system shall provide for the secure transfer of information from the FORSCOM WWMCCS system running at Top Secret system high to a set of WWMCCS Entry System (WES) users of a lower classification level. Special facilities provided by FSM will be controlled downgrading of information from FORSCOM WWMCCS to the WES sites, both manual and automatic, low user command filtering, extensive audit trail capabilities, and mechanisms to monitor and alter the FSM environment.

In order to accomplish the above FSM functions two FSM personnel types must be provided: the FSM Screener and the Environment Manager (EM). The Screener is responsible for insuring manually that information being downgraded (i.e., "written down") to low security level WES site users does not contain sensitive information of a higher classification. The Screeners perform their function via a Cathode Ray Tube (CRT) terminal which interfaces with FSM "trusted software." The trusted software will potentially be formally verified to guarantee that it does indeed operate correctly in accomplishing the downgrade. This special software guarantees that information is written down to the "low side" of FSM only if the Screener has viewed all the information and has approved and confir-

med a downgrade. The EM is responsible for a variety of tasks; the reassigning of low WES site users to a specific Screener, the monitoring of low WES site user activity, the establishment of special test/exercise parameters for automatic/manual screening, and the control of the FSM audit trail. The EM accomplishes his role via a terminal and untrusted software (albeit, some of the current functionality required for the exercise/experiment is potentially security relevant. However, these mechanisms would be removed in an operational environment).

Automatic screening and low user filtering will be accomplished with trusted software which interacts in concert with the Screener trusted software. Automatic screening is performed on those FORSCOM WWMCCS system outputs which are exactly recognized within the context of the user/system dialogue. Those system outputs which are not recognizable (i.e., termed "variable data") must be screened manually by an assigned Screener. Low WES site user filtering is accomplished by the same set of trusted software which performs automatic screening. For this situation the low user's input is restricted to a subset of the total FORSCOM WWMCCS capability.

The FSM functionality described above will be accomplished by a set of computer programs and data structures. The remainder of this section will describe these in detail.

### 3.1 PROGRAM IDENTIFICATION

A program is defined by the algorithms it requires and the data structures (i.e., the program "states") required by the algorithms. The specification and subsequent implementation of these algorithms is by the use of computer programs. These programs when activated are called processes. FSM is specified and implemented by a set of concurrent processes which operate over a set of data structures representing the current state of FSM. The remainder of this section presents and specifies in detail all FSM programs, data structures, and their interaction.

#### 3.1.1 Functional Areas

This subsection describes the basic functional areas which comprise the software of the FORSCOM Security Monitor (FSM). As discussed above, FSM will be implemented via a set of concurrent processes and a set of data structures which represent the state of FSM. Some processes will be invoked by user action (i.e., the Screener, the EM, or a low WES site user) while others may exist at all times performing required system functions. The data structures will be comprised of non-volatile disk files and volatile message queues. The precise specification of the states of these data structures in relation to the interaction of the processes constitutes the total algorithm of FSM operation. This section deals with the classes of FSM programs and descriptions of FSM data structures. Before discussing these in detail an overview of information flow in FSM will be given.

The overall information flow of FSM is shown in Figure 3-1. Here FSM programs are represented as executing processes (indicated as ellipses). Each program name is contained within the ellipse. The arrowed lines connecting processes indicate the direction of information flow, where in some cases the information flow is bi-directional (i.e., arrow heads on both ends). The FSM environment is divided into a high security domain and a low security domain. Note, except for the FSM Guardian Program (FSMGP), all FSM programs operate in the high security domain of FSM. FSMGP operates in both domains and has been given the privilege of violating, in a controlled manner, the \*-property of the DoD security model (i.e., FSMGP can write information to a lower security level). Consequently, FSMGP must be trusted software. FSM personnel are depicted at the top of Figure 3-1 and interface through the high security domain of FSM. This implies that they must be physically located in an area that is cleared Top Secret "system high". The WES site users are physically located at a site with a lower security classification. The FORSCOM WWMCCS complex is depicted at the bottom of Figure 3-1 and is also connected to the high security domain of FSM.

In order to illustrate the overall information flow in FSM a simple scenario is described below.



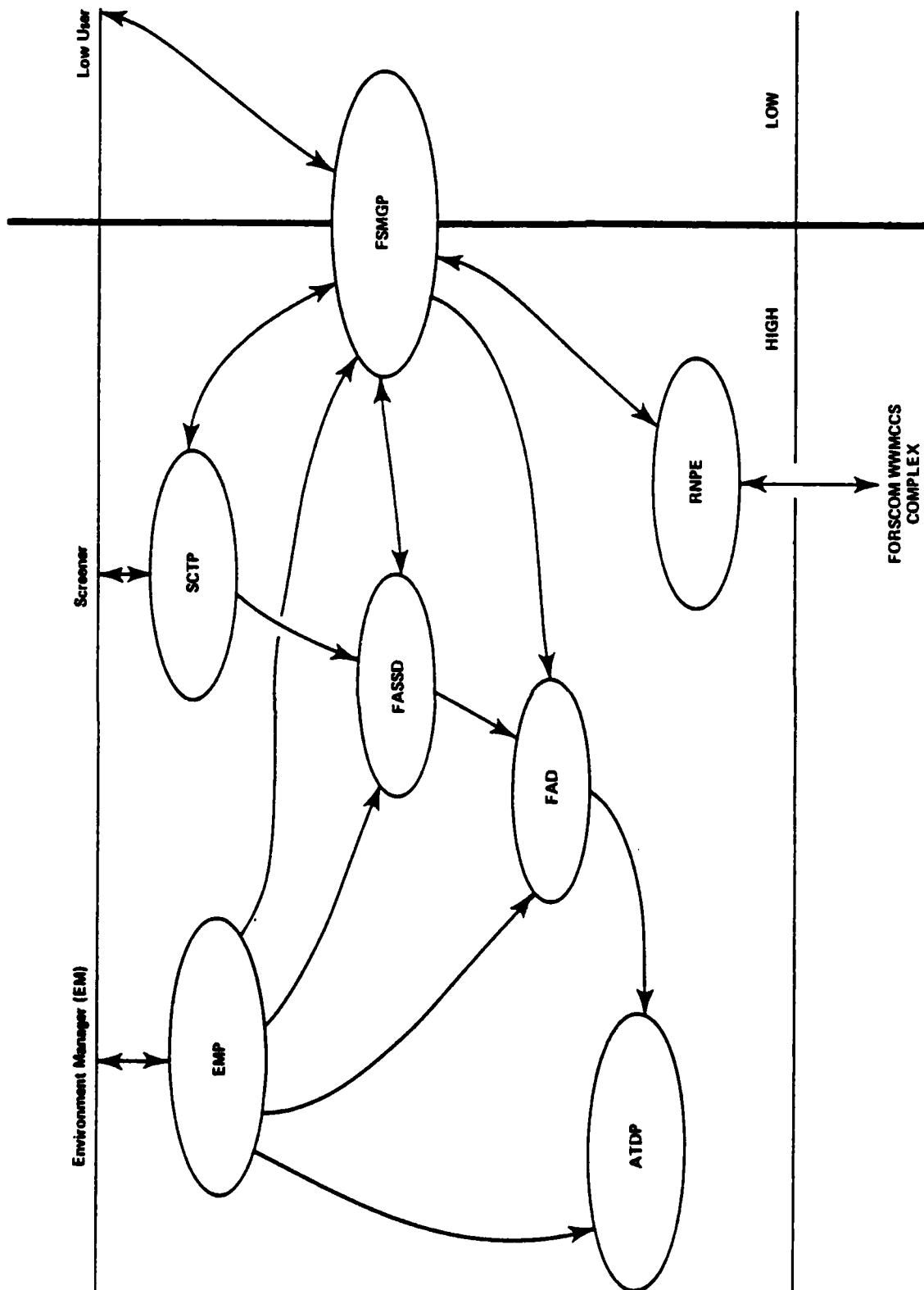


Figure 3-1. Basic FSM Process and Information Flow

For this scenario let's assume that one or more Screeners are currently logged on to the FSM system. Now a low WES site user wishes to access the FORSCOM WWMCCS system. The WES user logs onto the FSM via his terminal. The FSMGP is responsible for interfacing with the user (i.e., there will be a separate copy of FSMGP for each user interfacing to FSM). FSMGP requests a Screener assignment from the FSM Assignment Daemon (FASD) and allows the WES user to continue until a Screener is needed. Since Screeners are currently logged onto FSM one is available for assignment. Upon receiving the request from FSMGP, FASD reviews the set of FSM Screeners that are currently logged onto FSM. Based on their current load (i.e., other WES users may be logged onto FSM) and their logon time, FASD makes the appropriate Screener assignment. The Screener assignment is then passed back to FSMGP by FASD. Following logon to FSM, FSMGP logs the user on to the FORSCOM WWMCCS complex. This is accomplished by establishing a connection to WWMCCS via the Remote Network Process Emulator (RNPE). Following successful logon to WWMCCS, the WES user and WWMCCS continue with their dialogue, with FSM as a somewhat invisible intermediary. However, FSMGP is busily performing many internal tasks, one of which is auditing specific events. This is accomplished by writing out the audit event and required data which is periodically moved to the FSM audit trail by the FSM Audit Daemon (FAD). FAD is always present in the FSM environment and is responsible for recording other processes' audit events into the FSM audit trail. During the user/WWMCCS dialogue, FSMGP is constantly monitoring all data the user enters and all data the WWMCCS host is writing back to the user. If

the user inadvertently enters illegal command sequences (those which are either syntactically incorrect or are privileged and hence not available to the low user) FSMGP will inform the user of the error and not send the input to the WWMCCS system. This is the "filtering" mechanism of FSMGP. For WWMCCS system output, FSMGP determines if the data is recognizable (termed "fixed") or unrecognizable (termed "variable"). If the data is recognizable and the user's current screen mode permits it, FSMGP writes the information to the WES user's terminal. This is the "automatic screening" mechanism of FSMGP. If the data is not recognizable or if the user's current screen mode requires Screener intervention, FSMGP notifies the Screener Trusted Process (SCTP) that manual screening is required (remember, FASSD had assigned a Screener to this specific WES user). SCTP notifies the Screener (note, each Screener logged on to FSM has a separate copy of SCTP) of the "downgrade" request. The variable data from WWMCCS is then viewed, a screen at a time, for potential high level sensitive information. The Screener will screen all data and then make a decision whether to "downgrade" the information. Let's assume for this scenario, the Screener accepts the data. SCTP then asks the Screener to confirm his decision and following confirmation, notifies FSMGP that the data can be written down to the low user. FSMGP also audits this event which is then recorded into the FSM audit trail by FAD. The above sequence is the "manual screening" mechanism provided by FSM. This set of operations is continued until the WES user logs off FSM. FSMGP notifies FASSD of the logoff and then exits.

As a final comment on the overall information flow of FSM, EM can also peruse the FSM Audit Trail via the Audit Trail Display Program (ATDP). This program is called via the EMP as directed by the EM.

3.1.1.1 Programs. The FSM system consists of several different programs which are divided into two classes: transient programs and daemon programs. A transient program is normally brought into existence (i.e., becoming a process) via some external event. Following satisfaction of this event the process usually exits, hence the name "transient". Daemon programs are those programs which are usually brought into existence when the system is brought up and remain active as a process until the system is brought down. Also, daemons usually perform functions which are not tied directly to some external event (although almost always related indirectly to some external event) and appear "invisible", hence the name "daemon".

3.1.1.1.1 FSM Programs. Following is a list of FSM programs, each accompanied with a short description of its function.

a. FSM Guardian Program (FSMGP)

The FSM Guardian Program (FSMGP) interfaces directly with a low WES site user and is responsible for monitoring all information flowing from the WES user to the Top Secret WWMCCS system and vice versa. FSMGP contains the WES user "filtering" and "automatic screening" mechanisms of FSM. FSMGP operates over a set of extensive data structures

representing the user/system dialogue set for a WES user.

b. Screener Trusted Program (SCTP)

The Screener Trusted Program (SCTP) interfaces directly with a FSM Screener. The SCTP provides a verifiably secure mechanism by which a Screener can review information coming from the FORSCOM WWMCCS complex and accept/reject it for writing down to the low security level user. The process provides a "manual" screening facility for FSM.

c. FSM Assignment Daemon (FASSD)

The FSM Assignment Daemon (FASSD) is responsible for making initial Screener assignments to WES users. It also maintains overall FSM logon/logoff information in the Global Logon File (GLF).

d. FSM Audit Daemon (FAD)

The FSM Audit Daemon (FAD) operates invisibly within the FSM environment. It is responsible for maintaining the current FSM audit trail. It receives audit requests from the other processes of FSM and records these events onto the current FSM Audit Trail.

e. Environment Manager Program (EMP)

The Environment Manager Program (EMP) interfaces directly with the FSM Environment Manager (EM), performing specific tasks as directed by the EM. These tasks include reassigning FSM Screeners to WES site users, monitoring and modifying WES user environment parameters, and monitoring WES user activity profiles.

f. Audit Trail Display Program (ATDP)

The Audit Trail Display Program (ATDP) interfaces with the FSM Environment Manager (EM). This program allows the EM to selectively display portions of the FSM Audit Trail. Also, ATDP provides an audit trail maintenance facility by which the EM can save current FSM Audit Trails into other "time stamped" files.

g. Remote Network Processor Emulator (RNPE)

The Remote Network Processor Emulator (RNPE) is responsible for communicating with the FORSCOM WWMCCS complex. It emulates (by multiplexing/de-multiplexing WES users) a set of users interfacing to the WWMCCS complex.

3.1.1.1.2 Global Functions. This section provides descriptions of the global functions implemented for the FSM. Functions are defined in the literal sense as in the C language, hence are not programs but are actual functions which are compiled with the individual programs requiring their services. The global functions are described below.

3.1.1.1.2.1 Audit Function (GFAUDIT). To facilitate the logging of audit events, a global audit function is called by all event generators. The passed argument list is variable except that the first argument is always the one-byte event code (FEVENT). GFAUDIT processes its arguments and constructs an audit request file in the audit queue. It returns to the event generator when this is done. Note: The event generator does not wait for the event to be logged, only entered into the audit queue.

Input Parameters - Inputs to GFAUDIT consist of the passed arguments. With the exception of the first argument (which is always the audit event code, FEVENT), the argument list is of variable length and format. See Table 3-1 for the calling sequences to GFAUDIT, which are dependent upon the event code; see also Appendix D for detailed event definitions. These arguments are passed to GFAUDIT by any one of the four event generators (EMP, FASSD, FSMGP or SCTP):

- a. FEVENT — The audit event code.

This one-byte quantity ranges from 01 to 19 and indicates which event occurred. It is used by GFAUDIT to process the remaining arguments passed to it.

- b. SUBSYS -- The current WWMCCS subsystem code.

This one-byte quantity ranges from 1 to 8 and indicates the WWMCCS subsystem currently in use. A value of zero indicates all subsystems. See Table 3-2 for the subsystem names and their values.

- c. SRCFLG -- Text Source Flag

This flag indicates if the audit text pointer (TP) is a character pointer into a buffer (0) or if it is a character pointer to a filename (1) which contains the text to audit.

- d. TP -- The audit text pointer or filename where text can be found.

This character pointer is used to reference the beginning of the audit text (system output or user input). The event generator will set up this pointer and use a buffer for storage of the text. Alternatively, if there is a large amount of text the event generator will write a disk file instead of allocate in-core buffers. In this case, the SRCFLG will be set to one, not zero, as in the case of buffers. Upon return from GFAUDIT, the caller may de-allocate the buffer since the text is copied by GFAUDIT into the Audit Request File.



- e. FAILEX -- The coded reason why user input was not sent.

This one-byte quantity ranges from 1 to 2 and indicates the reason why user input was not sent to WWMCCS by FSMGP. A value of 1 indicates no match was made with the user input, while a value of 2 indicates a match was made but the input was disallowed. This value is only used in Event 02 (User Input Not Sent).

- f. SCRID -- The FSM user ID of the Screener

This one-byte quantity is identical, in concept, to the UNIX user ID and serves to uniquely identify the Screener involved in this event.

- g. STATUS -- The status of the subsystem initiation

This one-byte quantity indicates success (1) or failure (0) when a user attempts to initiate a WWMCCS subsystem. Note: Logon is not considered a valid WWMCCS subsystem in this context. This argument is only used in Event 11 (User Initiated WWMCCS Subsystem).

- h. USERID -- The FSM user ID of the WES user

Identical to the SCRID argument, except that it identifies the WES user involved in the event. A value of -1 indicates all users.

- i. USRTTY -- The WES user's UNIX terminal identifier

This one-byte quantity identifies the user terminal involved with this event. It is the last character of the user's UNIX terminal name ("/dev/tty?").

- j. SCRTTY -- The Screener's UNIX terminal identifier

This one-byte value is identical to the USRTTY argument, with the exception that it identifies the Screener terminal involved in this event.

- k. SCMODE -- The current screen mode

This one-byte quantity indicates whether no (0), normal (1) or all (2) screening is now in effect. It is only passed for Event 14 (Screen Mode Modified).

- l. RATE -- The bandwidth threshold for messages

This one-byte quantity indicates the number of messages which may be automatically downgraded without Screener intervention. It must be a positive integer in the range of one to 127.

- m. EMID -- The FSM user ID of the Environment Manager

This one-byte quantity is identical to the SCRID and USERID arguments, with the exception that it identifies the Environment Manager involved in this event.

n. EMTTY -- The EM's UNIX terminal identifier

This one-byte quantity is identical to the USRTTY and SCRTTY arguments, with the exception that it identifies the EM terminal involved in this event.

Output Parameters - None.

Operation - GFAUDIT assumes that the first argument passed to it is the one-byte event code (FEVENT). It uses this event code as a switch to decode the remaining arguments. After the arguments are extracted, GFAUDIT must obtain additional event data, such as the DTG of the event and its size. GFAUDIT then constructs an Audit Request filename from the event DTG, the process ID of the event generator (EMP, FASSD, FSMGP or SCTP) and a random character. This file is created in the Audit Request Queue and the event record is written as record one. If there is also text to audit (implies passing the TP and SRCFLG arguments), GFAUDIT copies the text into the second and possibly additional records. The text is copied "as is". It is null-terminated. GFAUDIT then closes the Audit Request File and returns control to the calling process.

Table 3-1. GFAUDIT Calling Sequences

Code	Description	Calling Sequence	Caller
01	User Input Sent to WWMCCS	gfaudit(01, userid, usrtty, srcflg, tp)	FSMGP
02	User Input not Sent to WWMCCS	gfaudit(02, userid, usrtty, subsys, failex, srcflg, tp)	FSMGP
03	System Output Sent to User	gfaudit(03, userid, usrtty, srcflg, tp)	FSMGP
04	System Output not Sent to User	gfaudit(04, userid, usrtty, subsys, srcflg, tp)	FSMGP
05	System Output Accepted by Screener	gfaudit(05, userid, usrtty, subsys, scrid, scrtty, srcflg, tp)	FSMGP
06	System Output Rejected by Screener	gfaudit(06, userid, usrtty, subsys, scrid, scrtty, srcflg, tp)	FSMGP
07	Screener Logged onto FSM	gfaudit(07, scrid, scrtty)	FASSD
08	Screener Logged off FSM	gfaudit(08, scrid, scrtty)	FASSD
09	WES User Logged onto FSM	gfaudit(09, userid, usrtty)	FSMGP
10	WES User Logged off FSM	gfaudit(10, userid, usrtty)	FASSD
11	User Initiated WWMCCS Subsystem	gfaudit(11, userid, usrtty, subsys, status)	FSMGP
12	User Assigned to Screener	gfaudit(12, userid, usrtty, scrid, scrtty)	FASSD
13	User Input Simulated	gfaudit(13, userid, usrtty, subsys, srcflg, tp)	FSMGP
14	Screen Mode Modified	gfaudit(14, subsys, userid, scmode)	EMP, FSMGP
15	Bandwidth Threshold Modified	gfaudit(15, userid, rate)	EMP
16	EM Logged onto FSM	gfaudit(16, emid, emtty)	EMP
17	EM Logged off FSM	gfaudit(17, emid, emtty)	EMP
18	System Output Received by FSMGP	gfaudit(18, userid, usrtty, subsys, srcflg, tp)	FSMGP
19	Downgrade Request Review Begun	gfaudit(19, scrid, scrtty, subsys)	SCTP

3.1.1.1.2.2 Date Conversion Function (GFDTGTOA). Input Parameters - GFDTGTOA requires a pointer to a buffer to store the converted ASCII time.

Output Parameters - GFDTGTOA outputs a pointer to the null character that terminates the converted ASCII string.

Operation - GFDTGTOA provides a common means for other processes to obtain a null-terminated ASCII string of the current Date-Time Group. GFDTGTOA uses a UNIX function to obtain the current date and then converts that to an ASCII string of the form MMDDHHmmss (Month, Day, Hours, minutes, seconds).

3.1.1.1.2.3 String Match Function (GFEQUAL). Input Parameters - pointers to two null-terminated strings to be tested for equality.

Output Parameters - GFEQUAL outputs a value of 1 if the strings are equal, zero otherwise.

Operation - GFEQUAL does a character-by-character comparison of the two strings until a mismatch occurs or the end of one is reached.

3.1.1.1.2.4 Error Function (GFERROR). Input Parameters - GFERROR requires one input parameter - an error/message number. GFERROR will accept any number of variable length null-terminated ASCII strings following the error/message number.

Output Parameters - None.

Operation - GFERROR provides a common means for the output of errors/messages to a user and system errors/messages to the system console. The first input parameter is an error/message number which specifies the error/message to be output. This number represents a canned message. If the number is greater than 500, then the error/message will be output to the system console and, in addition, if it is a system error and the standard output (file descriptor 1) is a terminal, a standard message will be output to the user indicating that a system error has occurred. Any further parameters will be inserted into the error/message at points where asterisks are encountered. Note: If GFERROR encounters a system error itself, it will output that error instead of the one with which it was called.

If the error is not a system error, the error message number indicates the appropriate error description item in GFEDES (error description file) which in turn contains pointers to the canned message text contained in GFEMES. Any character strings also provided as input parameters will be inserted in the canned message at points where asterisks are encountered. After printing the message to the user's terminal (or the system console for a system error/message), GFERROR returns to the calling function.

GFERROR uses library functions GFLOCK and GFUNLOCK to lock and unlock the system console in order to output system errors/messages. For this reason, it is mandatory that the calling program not have the system console locked when calling GFERROR.

GFERROR also uses two internal functions. These are:

- a. gfout - internal function which outputs a string.
- b. gfia - internal function which converts integer to ASCII.  
Input is an integer, output is a pointer to a null-terminated string in a fixed buffer (max of 9 characters).

3.1.1.1.2.5 Obtain User's Logon Name (GFGETNAME). Input Parameters -  
UNIX user ID.

Output Parameters - GFGETNAME returns a pointer to the user's logon name or to the strings "none" or "unknown", depending upon the user ID passed.

Operation - GFGETNAME opens the UNIX password file and searches for the passed user ID. If found, GFGETNAME returns a pointer to the user's logon name. Otherwise, GFGETNAME returns a pointer to the string "unknown". If no user ID is passed to GFGETNAME, it returns a pointer to the string "none".

3.1.1.1.2.6 File Copy Function (GFILECOPY). Input Parameters -  
GFILECOPY requires three input parameters:

- a. A flag with the following values:
  - 0 - Create destination file.
  - 1 - Append data to existing destination file.
- b. Pathname of source file.
- c. Pathname of destination file.

Output Parameters - GFILECOPY returns a null if the copy was successful, or a positive system error number (with which to call GFERROR) if unsuccessful.

Operation - GFILECOPY provides a mechanism for the copying of one file to another. The first input parameter is a flag which specifies whether the destination file is to be created or not. If the flag is set, the destination file is assumed to exist, and if it cannot be opened, GFILECOPY will return with an error (Open error). The second input parameter is the pathname of the source file, and the third input parameter is the pathname of the destination file. GFILECOPY first attempts to open the source file. If unsuccessful, then it returns an open error to the calling function. Otherwise, GFILECOPY attempts to open/create the destination file, depending on the value of the input flag. If the action is unsuccessful, GFILECOPY returns an open/create error to the calling process. If append is indicated, GFILECOPY seeks to the end of the destination file before beginning the copy. It then reads blocks of data from the source file and writes them to the destination file until the End-Of-File (EOF) of the source file is reached. If a read or write error is encountered at any time, GFILECOPY returns that error to the calling function. Otherwise, after a successful copy, both files are closed and GFILECOPY returns a successful value to the calling function.

3.1.1.1.2.7 Integer to ASCII Function (GFITOS). Input Parameters - an integer to be converted to an ASCII string.

Output Parameters - GFITOS returns a pointer to the beginning of the



ASCII string.

Operation - GFITOS allocates space for the ASCII string and then uses a modulo operation to fill the buffer from the end to the beginning. If the input integer was less than zero, GFITOS prepends a minus sign character to the ASCII string.

3.1.1.1.2.8 Lock Function (GFLOCK). Input Parameters - GFLOCK requires one input parameter - a pathname of the file to be locked.

Output Parameters - None.

Operation - GFLOCK provides a common means for the locking of data files which must be updated by more than one process. GFLOCK first builds a "lock-file" pathname by concatenating the string "-lock" to the supplied pathname. GFLOCK then attempts to create the lock-file with a read-only mode. If the file already exists, an error status is received. GFLOCK then sleeps for one second and then attempts the create again. Eventually, the create is successful and GFLOCK returns to the calling function. At this point, the file is "locked-out" from other read-for-update attempts. Of course, this assumes the cooperation of other processes which update this file to call GFLOCK prior to such updates. The file can now be updated, and after the update is complete it is mandatory that the calling process call the GFUNLOCK function to "unlock" the file. Note: The user ID of the calling process must not be that of the superuser since, in that case, the create is always successful even if the file does already exist.

3.1.1.1.2.9 Leading Zero Integer to ASCII Conversion Function (GFLZITOA). Input Parameters - GFLZITOA requires three parameters:

- a. Number of digits to convert to a string.
- b. Integer value to convert.
- c. A pointer to a user supplied buffer to place the converted string in.

Output Parameters - None.

Operation - GFLZITOA first converts the integer to an ASCII string, using the same technique as does GFITOS. It then checks the leading characters of the ASCII string and if they are blank, GFLZITOA converts them to character zeroes. Finally, the ASCII string is stored character-by-character beginning at the buffer location given by the third argument.

3.1.1.1.2.10 List Match Function (GFMATCHLIST). Input Parameters - a pointer to an array of strings and a pointer to a string to match to the array.

Output Parameters - GFMATCHLIST returns the index of the array string matched, if found. Otherwise it returns a -1 for no match or a -2 if a match was made to more than one array string.

Operation - GFMATCHLIST first makes certain both array and search string are lower case, and then searches the input array for any occurrences of the input string. If only one exact match is made, the index of the

matched array string is returned. Otherwise a -1 is returned if no match is found or a -2 is returned if multiple array strings matched the input string.

3.1.1.1.2.11 String Match Function (GFNEQUAL). Input Parameters - GFNEQUAL requires three input parameters:

- a. Length of match.
- b. Pointer to a string to match.
- c. Pointer to a second string to match with.

Output Parameters - GFNEQUAL returns a 1 if both strings are equivalent for the amount of characters specified, otherwise a zero is returned.

Operation - GFNEQUAL compares the two strings character-by-character until a mismatch is detected or the specified number of bytes have been checked, whichever occurs first.

3.1.1.1.2.12 Read IPC Function (GFRIPC). Input Parameters - GFRIPC requires three input parameters:

- a. A file descriptor for the port.
- b. A pointer to a user supplied buffer in which to store an IPC message.
- c. A flag with the following value:
  - 0 - Block on port read.
  - 1 - Poll on port read.

2 - Block on port read, but return EOF if all writers die.

Output Parameters - GFRIPC returns the number of bytes in the current IPC message, 0 for EOF, or -1 if a read error has occurred.

Operation - GFRIPC provides a common means for the reading of ports. The first input parameter is a file descriptor which the user obtained from a successful port call. The second input parameter is the location of a buffer into which the input IPC message will be placed. The third input parameter is a flag with three values. The value of zero (0) indicates that the caller wishes to block on port reads, i.e., does not want GFRIPC to return until a message has been read. The value of one (1) indicates that the caller does not want to block on reads, but wants GFRIPC to return EOF if there is no data available. The value of two (2) indicates that the caller wishes to block on port reads, but does want GFRIPC to return EOF if all of its writers have died. GFRIPC returns one of three values. A count is returned if an IPC message has been read. An EOF (0) is returned if the user has indicated that polling is desired and there is no data, or if the user has indicated that partial blocking is desired and all writers have died. A minus one (-1) is returned if GFRIPC encounters a read error from the port.

GFRIPC uses the alloc and free system calls in order to buffer up IPC messages from more than one writer.

GFRIPC first checks if polling is wanted, and if so performs an EMPTY call on the supplied file descriptor. If no data is available, GFRIPC returns a no data value to the calling function. Otherwise, it reads on the supplied file descriptor. If a system error occurs (read error), a read error value is returned to the calling function. If an End-Of-File (EOF) is encountered (all writers have died) and partial blocking was requested, a no data value is returned to the calling function. Otherwise, GFRIPC reads an IPC header. If this is the first block for this message (indicated by the Process ID (PID) in the header), then GFRIPC allocates enough core for an entire message. If this is not the first segment of a message, then GFRIPC searches for the start of the message in already allocated core. It then reads the block of data into the allocated core. If this is the end of a message, then GFRIPC moves the entire message into the user supplied buffer, deallocates the core and returns the number of characters in the message to the calling function. Otherwise, it reads again on the port for another message segment.

3.1.1.1.2.13 Compute String Size Function (GFSIZE). Input Parameters - a pointer to a null-terminated string whose size is to be computed.

Output Parameters - the number of characters in the string.

Operation - GFSIZE steps through the string counting characters until it finds a null character. It then returns the character count to the caller. Note: the terminating null byte is not included in the count.

3.1.1.1.2.14 String Copy Function (GFSTRCOPY). Input Parameters - a pointer to a null-terminated string to be copied and a pointer to a buffer in which to place the copy.

Output Parameters - a pointer to the terminating null byte in the destination buffer.

Operation - GFSTRCOPY copies the input string character-by-character into the destination buffer until it reaches the terminating null character in the source string.

3.1.1.1.2.15 Unlock Function (GFUNLOCK). Input Parameters - GFUNLOCK requires one input parameter - a pathname of the file to be unlocked.

Output Parameters - None.

Operation - GFUNLOCK provides a common means for the unlocking of data files which must be updated by more than one process. GFUNLOCK first builds a "lock-file" pathname by concatenating the string "-lock" to the supplied pathname. GFUNLOCK then unlinks the lock-file, thus allowing another process to create it and hence gaining control of the file. Of course, failure of a process to call GFUNLOCK after successfully locking out a file with GFLOCK would eventually bring the FSM to a halt.

3.1.1.1.2.16 Write IPC Function (GFWIPC). Input Parameters - GFWIPC requires three input parameters. It will accept a variable number of other parameters as specified below.

- 1
- a. A flag specifying interpretation of the second parameter:
    - 0 - second parameter is a pathname of a port to open for write,
    - 1 - second parameter is a file descriptor of a port already opened.
  - b. Pathname of a port, or a file descriptor depending on the value of the first parameter.
  - c. IPC message type as defined in the FSM global data.
  - d. Optional parameters, in the following order if present: User ID, Bandwidth rate, Screen Mode, Subsystem index, user security classification, pathname of a downgrade file, pathname of a context file.

Output Parameters - GFWIPC returns the status of the write: 0 - successful, positive error number with which to call GFERROR if write was unsuccessful.

Operation - GFWIPC provides a common means for writing to ports. The first input parameter is a flag which tells GFWIPC whether the second parameter is a pathname of a port to open or a file descriptor of an already opened port. The second input parameter is then either a file descriptor or a port pathname. The third input parameter is an IPC message type which will be stored in the IPC port message structure. Any other parameters are data to be stored in the IPC port message structure.

GFWIPC sets up the IPC message in the structure according to the input parameters which have been passed. If a pathname was input, then GFWIPC opens the port. If the open is unsuccessful, an open error is returned to the calling function. Otherwise, GFWIPC writes the IPC message to the opened port. If a write error occurs, then GFWIPC returns a write error to the calling function. If a pathname was input, GFWIPC closes the port. It then returns to the calling function with a successful value.

3.1.1.2 Data Structures. The state of FSM is contained in a set of data structures which can be divided into two types:

- a. non-volatile disk files
- b. volatile message queues (i.e., ports)

Disk files are used to contain that information which must be saved through continuous day-in and day-out operations of FSM. Audit trail information is a good example of information contained in disk files. In addition other system oriented information must be maintained in disk files in order for FSM to operate correctly; i.e., predefined data which represents the user/WWMCCS dialogue contexts. Volatile data is that data which is not needed in FSM for long periods of time; message queues for inter-process communication (IPC) are good examples of volatile data. All message queues will be implemented in FSM via the port data structure. Ports provide sophisticated First-In, First-Out (FIFO) queues for efficient inter-process communication between processes. In addition, ports allow inter-process communication across different process fami-



lies, hence allow transient/daemon process communication.

Figure 3-2 illustrates the entire FSM directory/data structure hierarchy. Before describing this structure the notational conventions used in this figure will be defined.

- a. Rectangles represent files or directories with the file/directory name appearing on top. The name is followed by "(d)" if it represents a UNIX directory, and "(f)" if it is a disk file.
- b. Disk files entered in a directory have their names inside the rectangle representing their presence.
- c. Circles represent ports, with the port name appearing inside the circle.
- d. UNIX C-structure names appear in capital letters (e.g., GPROF).
- e. Special lock-files are denoted as dashed-line entries in directories (e.g., gaud-lock).
- f. Symbolic names are employed to depict multiple occurrences of a name. For example "user-1" through "user-n" is used to depict different FSM WES users.



To determine the entire UNIX pathname of any given data structure, each directory-to-directory/file/port must be concatenated together. For example, the pathname for the profile file of WES user-i is `"/fsm/usr/user-i/gprof"`. Similarly, the absolute pathname for the Assignment Daemon (FASSD) input port is `"/fsm/port/faport"`.

Before going into the detailed descriptions of the individual data structures depicted in Figure 3-2 a general overview will be given on the entire directory/data structure hierarchy.

All directory/data structures originate directly or indirectly from the UNIX "root" directory, termed `"/"`. FSM will have its own directory with the absolute pathname of `"/fsm"`. It contains system level data structure entries and other related directories. System level data structures are the user logon file (`"/fsm/glf"`) which contains information on all FSM personnel and WES users currently logged onto FSM; the system level environment file (`"/fsm/genv"`) which contains information currently representing system-wide FSM environment values; the user/system dialog context structures for teletypewriter (`"/fsm/gcont.tty"`, `"/fsm/ghdel.tty"` and `"/fsm/gtext.tty"`) and VIP (`"/fsm/gcont.vip"`, `"/fsm/ghdel.vip"` and `"/fsm/gtext.vip"`) terminals which control the automatic output screening and input filtering mechanisms of FSM; the error description file (`"/fsm/gedes"`) which contains a pointer to the first character of an error message text and number of characters in the message (used by GFER-ROR); the error messages themselves (`"/fsm/gemes"`); and finally, the global terminal file (`"/fsm/gttys"`) which is used by the FSMGTTY program to

start up a Guardian Process for specified WES terminals.

The FSM directory contains three directories relevant to data structures: the low user directory ("/fsm/usr"), the FSM audit trail directory ("/fsm/fsat") and the IPC port directory ("/fsm/port"). The user directory contains entries to individual directories for each WES user logged on to FSM. Individual directories exist regardless of whether the user is logged on or not. For user-i, the directory "/fsm/usr/user-i" would contain four data structure references; the user profile file ("/fsm/usr/user-i/gprof"), the user environment file ("/fsm/usr/user-i/genv"), the downgrade file ("/fsm/usr/user-i/gdown") and the user downgrade context information file ("/fsm/usr/user-i/ginfo"). The FSM audit trail directory, "/fsm/fsat", contains the entire FSM audit trail environment. This directory resides on a separate file system to ensure the security of the generated audit trail, since it will contain Top Secret data. The directory contains one directory and two classes of audit trail events/texts, and one lock-file. The primary audit file data structure is "/fsm/fsat/gaud" which contains information describing the current state of the FSM audit trail. Directory "/fsm/fsat/auditq" is used by FSM processes for registering audit trail events. For example, the FSM Guardian Program (FSMGP) in auditing an event would insert an entry into this directory using the current Date-Time Group (DTG), its process identifier (PID) and a random character as a file name. This combination ensures uniqueness within the directory "/fsm/fsat/auditq". Data structure "/fsm/fsat/current.e" and "/fsm/fsat/current.t" contain the event and text information respectively of the current audit trail.

Similarly, data structure entries `"/fsm/fsat/dtg.ei"` and `"/fsm/fsat/dtg.ti"` contain previously saved audit trails and are identified by a unique DTG. Finally, the lock-file, `"/fsm/fsat/aud-lock"` is used when the audit trail file is being updated (i.e., the file `"/fsm/fsat/aud"`).

The last directory in the main FSM directory is the port directory (`"/fsm/port"`). This directory contains all FSM ports. FSM utilizes five basic ports, the FSM Assignment Daemon (FASDD) port (`"/fsm/port/faport"`), the FSM Screener assignment port (`"/fsm/port/fsagn-i"`), the Screener request port (`"/fsm/port/screq-i"`), Screener reply port (`"/fsm/port/fsrly-i"`) and the Screener logoff port (`"/fsm/port/sclog-i"`). The fsagn and fsrly ports exist for each logged on WES user, while the screq and sclog ports exist for each Screener that is logged on. These four portnames are constructed for uniqueness using the owner's FSM user ID (hence the `"-i"`). All ports only exist when the reading process is activated within FSM. Note, since FASDD is a daemon the faport is always present.

3.1.1.2.1 Disk File Descriptions. Each disk file that appears in Figure 3-2 will be described in detail in the following sections.

3.1.1.2.1.1 GCONT (Global Context File). This structure provides the information FSMGP needs to follow a user's progress through the various WWMCCS subsystems allowed in the restricted environment. It is tree-like in its structure, since any given item will reference several other items in the file. The file contains a series of CONTEXT items. A CONTEXT

item (termed "CONTEXT") represents all possible recognizable user inputs and/or system outputs at a branch of the input/output dialog tree.

Each CONTEXT item contains a HEADER item and one or more SUBCONTEXT items. A HEADER and SUBCONTEXT define a CONTEXT. See Figure 3-3 for a depiction of their relationship. The HEADER supplies information that is applicable to the entire CONTEXT while SUBCONTEXTs supply information relevant to a given user input or system output. The HEADER allows FSMGP to know when to stop accumulating system output or user input. It also provides default actions on error conditions. Figure 3-4 shows a HEADER item. A CONTEXT can have one or more SUBCONTEXTs, depending on the possible recognizable user inputs/system outputs at this stage of the user/system dialog. All SUBCONTEXTs, whether user input or system output are the same size. See Figure 3-5 for a SUBCONTEXT item description.

#### HEADER Field Descriptions

**GPOPTR** The pop-up context pointer points to a recovery CONTEXT item when FSMGP encounters an error condition or a user-entered break. Error conditions include a no match condition between current SUBCONTEXT items and system output, a time-out condition on system reads, and Screener rejection of system output. FSMGP usually performs some action (i.e. simulate user input) to resynchronize its processing with the next system output and then proceeds to the appropriate CONTEXT. GPOPTR points to this item, in general a higher level within a given WWMCCS subsystem.

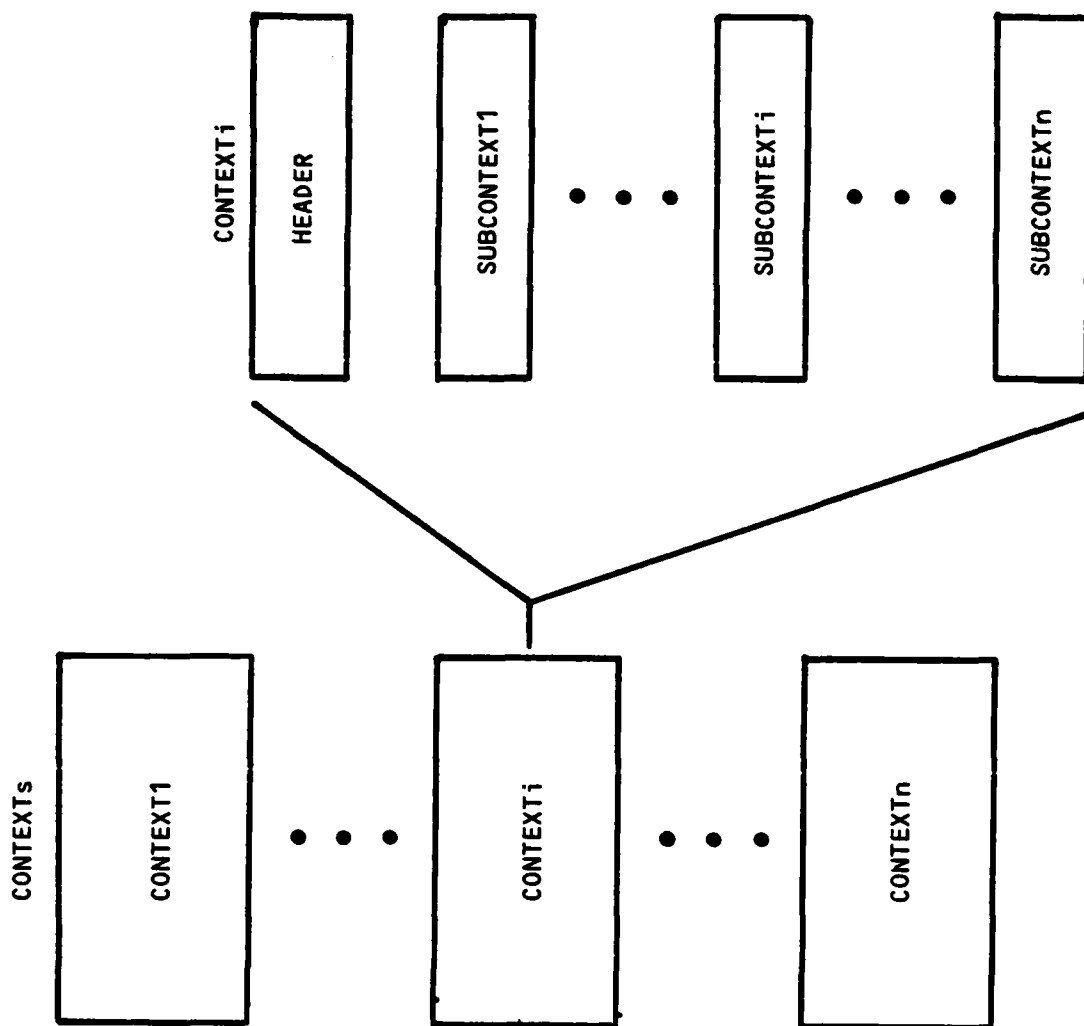


Figure 3-3. Header and Subcontext Relationship

Word 0	GPOPTR	
Word 1	GDLIMP	
Word 2	GNSUBS	GCONFL
Word 3	GFUNX	GSYSX

BIT0	BIT1	BIT2	BIT3	BIT4	BIT5	BIT6	BIT7
			BOTH	USR	RPT	FIX	AUD

GCONFL (from above)

Figure 3-4. GCON Header Item



Word 0	GSTRP	
Word 1	GNEXT	
Word 2	GSFX	GSUBFL
Word 3	GSTART	
Word 4	GLENG	

BIT0	BIT1	BIT2	BIT3	BIT4	BIT5	BIT6	BIT7
		FX	SUB	FUNC	VAL	USI	BEG

GSUBFL (from above)

Figure 3-5. GCON Subcontext Item

**GDLIMP** The delimiter set pointer points to a byte offset in the Halting Delimiter file at which the particular delimiter set for this CONTEXT resides. FSMGP reads user input or system output until one of these strings is matched.

**GNSUBS** The number of subcontexts field specifies the number of SUBCONTEXTs comprising this CONTEXT. Since SUBCONTEXTs are of a fixed size, this value can be used in reading the complete CONTEXT for processing by FSMGP.

**GCONFL** The header flags field contains various bit settings representing conditions of the CONTEXT as a whole, in contrast to states specific to individual SUBCONTEXTs. The following bit settings are used:

**BOTH** User Input and System Output (1). This bit, when set, signifies that either user input or system output can be expected next. When this bit is set, FSMGP must poll both the user and the system for the next data. Note: when this bit is set, the USR bit is ignored.

**USR** User input(1) / System Output (0). This bit signifies whether the CONTEXT represents user input or system output. It is only used when the BOTH bit is zero.

**RPT** Repeating CONTEXT (1) / Non-repeating CONTEXT (0). This bit signifies whether the CONTEXT requires multiple iterations of the FSMGP Read System Output (RSO) function in order to satisfy a single SUBCONTEXT.

**FIX** Fixed Repeating CONTEXT (1) / Variable Repeating CONTEXT

(0). Only applicable when the RPT bit is set to one. In either situation of this bit setting, multiple calls to the RSO function are required in order to satisfy this CONTEXT. For fixed CONTEXTs, FSMGP will call the Read System Output function repeatedly, doing a continuous match on the indicated search string specified by the single SUBCONTEXT. This will allow, for example, a simple delimiter string set for halting by the RSO function (probably line-by-line for some lengthy output; e.g., TCON skeletons). For variable CONTEXTs, FSMGP will accumulate the variable data until a match is made by one of the SUBCONTEXTs. This mechanism is applicable, for example, to TLCF message texts.

AUD Audit subsystem entry (1) / no audit required (0). This bit signifies that an audit trail event is required because the user just entered another WWMCCS subsystem. The subsystem entered is specified in field GSYSX (see below).

GFUNX This entry indexes a function in FSMGP which will generate the appropriate simulated user input required to place FSMGP (and the WES user) back to the "pop-up" level as specified in field GPOPTR. Different sequences are required for different subsystems.

GSYSX This field specifies which subsystem the CONTEXT is concerned with. It will be filled with an index that identifies the subsystem name. See Table 3-2 for subsystem names and their values.

## SUBCONTEXT Field Descriptions

- GSTRP** This field is an offset into the GTEXT data structure which specifies the string to be searched against for this particular SUBCONTEXT. The SUB bit field in GSUBFL controls how this search is performed.
- GNEXT** This field is an offset into GCONT specifying the next valid CONTEXT to transfer to if this SUBCONTEXT satisfies the given user input/system output.
- GSFX** This field indexes a function in FSMGP and indicates that a special function is required to validate user input parameters or to modify system output.
- GSUBFL** This field contains a set of bit flags which represent various conditions of this particular SUBCONTEXT. The following bits are defined:
- FX** Fixed data (1) / Variable data (0). This bit in conjunction with the SUB bit indicates the type of match algorithm to use. Fixed data (whether user input or system output) has two cases: substring and no substring. Fixed, no substring matching is perhaps the easiest to comprehend since it involves a character-by-character comparison between the input data and the text string in GTEXT. Fixed, substring on the other hand utilizes two additional values (GSTART and GLENG) to define a window, or substring, within the GTEXT text string to match against. This window is compared against the input data and if a match occurs, the

GTEXT string is written to the user, not the input data. If this bit is not set (i.e. variable), again there are two cases to consider: substring and no substring. Variable, substring is very similar to fixed, substring with the exception that the text string in GTEXT is not further defined and is assumed to be the substring with which to match. Variable, no substring (otherwise known as a "null subcontext") is a wildcard and will match any input data. If the screen mode is normal, fixed data is a candidate for automatic screening by FSMGP, while variable data must be visually inspected by the Screener.

- SUB Input data requires Substring match (1) / no substring match (0). Used in conjunction with the FX bit, see description given above.
- FUNC This bit signifies that a special function is required (1) / function not required (0). Field GSFX contains the index of the special function if this bit is set.
- VAL This bit signifies that a user's matched input is invalid (1) / matched input is valid (0). This provides a finer granularity on the filtering of user input.
- USI This bit signifies the subcontext as user input (1) / system output (0). It is used only if the BOTH bit flag in the CONTEXT header is set.
- BEG This bit signifies that the halting delimiter for the subcontext is the beginning of the next subcontext (1) / the

end of the current subcontext (0). Since system output strings may terminate with unmatchable variable data, a given halting delimiter may apply to the beginning of the string expected next.

**GSTART** This field is a byte offset into a fixed system output at which to begin the subcontext match. This field is relevant for fixed, substring system outputs only.

**GLENG** This field specifies the length of the system output to be matched. It is used in conjunction with GSTART to make an exact match on the substring at the specified location within the system output. This field, like GSTART, is relevant for fixed, substring system outputs only.

**3.1.1.2.1.2 GTEXT (Global Text File).** This structure contains the actual text strings of user input and system output. These text strings are used by FSMGP when it is identifying the user/system context. When FSMGP has obtained a user input or system output, it must identify it to recognize the user's place in the particular WWMCCS subsystem. This is accomplished by comparing the system output or user input with a subcontext string. Strings in GTEXT are referenced by the GSTRP field in the SUBCONTEXT item. GTEXT is thus simply a concatenation of all possible user input/system output strings in one file, each terminated by a null character. When a particular SUBCONTEXT is checked to see if it matches a particular system output or user input, the SUBCONTEXT field GSTRP points to the appropriate string to match. FSMGP then obtains that string and makes a comparison.

3.1.1.2.1.3 GHDEL (Global Halting Delimiter File). This structure contains all the character strings required by FSMGP to determine when it has accumulated enough system output or user input to satisfy a particular CONTEXT. These delimiter strings are arranged in sets, since the SUBCONTEXTS of a given CONTEXT may be terminated differently. Each delimiter string in the file is terminated by a null character, and the final string in each set of delimiter strings is terminated by two null characters. This allows FSMGP to know when it has reached the end of a particular set.

FSMGP must read system output and user input, but in order for it to know when enough data has been read (and therefore to stop) it must be aware of the possible terminating sequences of any given CONTEXT. GHDEL provides this information. After a new CONTEXT is established by FSMGP (pointed to by the old one), it must read data (system or user) to find a match to one of the particular SUBCONTEXT strings in the current CONTEXT. A read will continue until one of the delimiter strings is found (or an error occurs). The particular string set that FSMGP searches against is pointed to by the GDLIMP field in the current CONTEXT's HEADER.

3.1.1.2.1.4 GDOWN (Global Downgrade File). This structure contains the text of the system output that is to be examined by the Screener. It is created by FSMGP and written to each time a downgrade is required. When a system output is recognized as requiring screening (either because it is variable data or because the current screen mode is set for all data to be screened), FSMGP sends an IPC message to SCTP requesting a down-

grade. In this message is the pathname of the user's GDOWN (uniquely identified by user) which SCTP displays to the Screener to accept or reject.

3.1.1.2.1.5 GINFO (Global Downgrade Context File). This structure contains the latest user/system interactions so that the Screener may make his downgrade decision in concert with the current user/system dialog. It contains recent user input and system output text strings in the order of their occurrence. Whenever a downgrade is required FSMGP writes the data from the Downgrade Context (GINFO) Buffer to this file. The Screener may peruse this file if he/she desires.

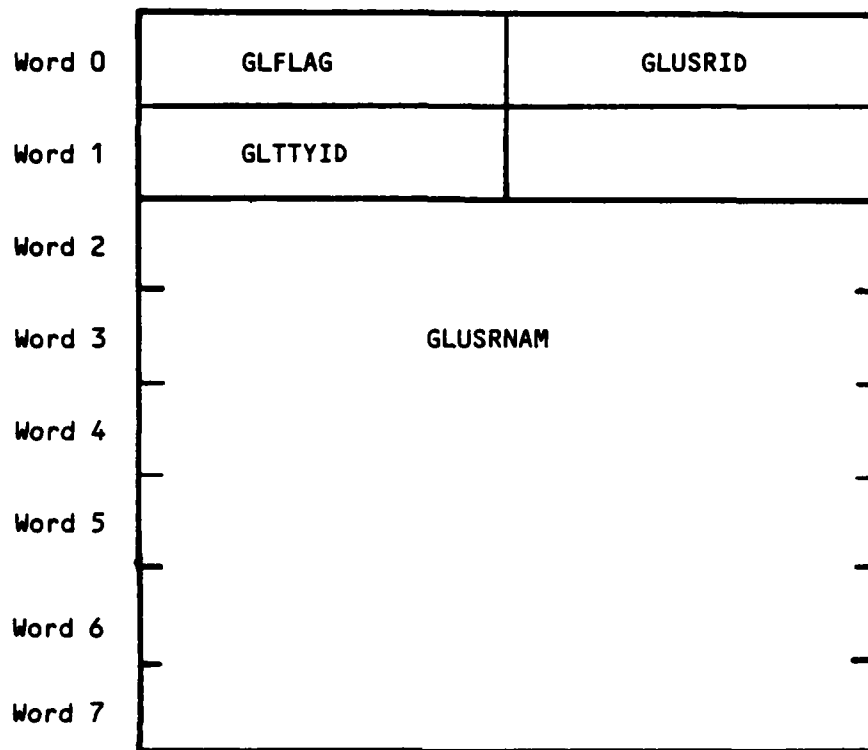
3.1.1.2.1.6 Global Logon File (GLF). The Global Logon File is established at system generation time and contains one record for each FSM user. The record contains user identification (in the form of FSM logon name) and bit-flags denoting if the user is a Screener, an EM or a WES user and if the user is currently logged on. A user's FSM user ID is obtained by noting the location of the user's record in the file. For example, the user described by record 14 has a FSM user ID of 14. In this way, the normal UNIX user ID is not used. Since the FSM user ID is a one-byte quantity, GLF must contain no more than 127 records (the zeroth record is not used). This file will be updated by the Guardian Process (FSMGP), FSM Assignment Daemon (FASD), Environment Manager Process (EMP), Screener Trusted Process (SCTP) and by a stand-alone program USER (see Appendix E). In light of the number of potential writers, GLF has an associated lock file ("/fsm/glf-lock") which must be created prior



to writing. When a user logs onto the FSM, the user's process (whether EMP, SCTP or FSMGP) checks this file for a record containing the user's name. If there is no match found or a match is made and either the DEFINED bit is not set or the user-type bit (DHI-EM, DHI-SCR or DLO-USR) is not proper, the user cannot proceed and his process is terminated. If a record is found and all bit settings are correct, the user's process marks the record active by setting the DLOGDON bit to one. When a Screener or WES user logs off FSM, FASSD is notified and resets the user's terminal ID to zero and marks the record inactive by setting the DLOGDON bit to zero. EM logoff is taken care of by the EMP. Whenever FASSD makes a WES user/Screener assignment, the WES user's record is modified to contain the assigned Screener's FSM user ID. Finally, when the EM manually reassigns a WES user to a Screener, FASSD modifies GLF. EMP reads this file when processing its commands. See Figure 3-6 for a record layout of the GLF file. The corresponding fields are described below:

LFLAG A collection of bit flags that have the following meanings:

DEFINED This bit signifies that the user which this record describes is a valid FSM user. In this way, records are never removed from GLF (thus altering a user's FSM user ID), but are marked as not-DEFINED, when a user is deleted from FSM participation.



BIT0	BIT1	BIT2	BIT3	BIT4	BIT5	BIT6	BIT7
			DE- FINED	DLOG- DON	DHI- EM	DHI- SCR	DLO- USR

LFLAG (from above)

Figure 3-6. A GLF Record

DLOGDON This bit signifies that this user is currently logged onto FSM. During logon, this bit is checked and set by the user's process (EMP, SCTP or FSMGP). During logoff, FASSD resets the bit for WES users and Screeners, while EMP resets it for the EM. Its purpose is to prevent users with the same name from being logged on simultaneously.

DHI-EM This bit indicates that this GLF record describes a "high-side" Environment Manager. Only one record should be marked as an EM.

DHI-SCR This bit indicates that this GLF record describes a "high-side" Screener.

DLO-USR This bit indicates that this GLF record describes a "low-side" WES user.

LUSRID The FSM user ID of this WES user's currently assigned Screener. If there is no Screener currently assigned, the field will be zero. This field is undefined (not used) for Screener and EM records.

LTTYID The UNIX identifier of the terminal this user is logged onto or zero if the user is not logged on.

LUSRNAM The eight-byte UNIX logon name of this Screener or EM; or the twelve-byte WWMCCS user ID of this WES user.

3.1.1.2.1.7 Global Logon File Lock (GLF-LOCK). Because FSMGP, FASSD, SCTP, EMP and USER all modify the Global Logon File (GLF), the standard lock mechanism is used during writes to GLF. The existence of the lock file indicates that a process is currently updating GLF. The lock file has no records, it is merely an entry in the /fsm directory so that its presence or absence may be easily checked.

3.1.1.2.1.8 System Global Environment (GENV) File. Established at FSM initialization is the System Global Environment File. When a WES user logs onto FSM, he is given a copy of the system default values. Subsequent changes in the environment can be made by either the EMP, at the behest of the EM, or by the FSMGP, following detection of bandwidth overflow. The file contains one record with values which indicate bandwidth threshold and subsystem screen modes. Updates occur when the EM, via the EMP, modifies system-wide or user-specific environment values. See Figure 3-7 for a GENV record layout. The corresponding fields are described below:

BANDW The bandwidth threshold. Indicates number of messages per minute which are allowed to be automatically downgraded.

ACCSM The ACCESS subsystem screen mode. A screen mode has one of three discrete values: OFF (no screening is done), NORMAL (FSMGP determines what system outputs need to be sent to the Screener for manual downgrading) or ALL (all output is manually screened).

LSTSM The LIST subsystem screen mode. Screen mode can be OFF (0), NORMAL (1) or ALL (2).

Word 0	BANDW	ACCSM
Word 1	LSTSM	SIOSM
Word 2	TCNSM	TLCSM
Word 3	TSSSM	WDMSM

Figure 3-7. A GENV Record

SIOSM The SIOS subsystem screen mode. Screen mode can be OFF (0), NORMAL (1) or ALL (2).

TCNSM The TCON subsystem screen mode. Screen mode can be OFF (0), NORMAL (1) or ALL (2).

TLCSM The TLCF subsystem screen mode. Screen mode can be OFF (0), NORMAL (1) or ALL (2).

TSSSM The TSS subsystem screen mode. Screen mode can be OFF (0), NORMAL (1) or ALL (2).

WDMSM The WWDMS subsystem screen mode. Screen mode can be OFF (0), NORMAL (1) or ALL (2).

3.1.1.2.1.9 User Global Environment (GENV) File. Associated with every logged-on WES user is a GENV file which contains environment values for the specific user. The file is created when the user logs onto FSM. The user's FSMGP copies the system GENV into the user's GENV. The file is contained in the user's directory ("/fsm/usr/user-i"). The file is modified, when environment values change, by FSMGP. The format of the file follows the system GENV exactly (refer to Figure 3-7). FSMGP modifies this file when the EM makes environment changes pertinent to this WES user and when bandrate overflow has been detected. EMP reads this file when displaying a current WES user's environment to the EM.

3.1.1.2.1.10 Global Profile (GPROF) File. Associated with each WES user is a user profile file. This file is created and updated by the user's FSMGP. The file contains only one record which identifies the user (with FSM user ID and UNIX terminal ID), indicates the current WWMCCS subsys-

tem, user's status and the Date-Time Group (DTG) of the status. Status in this sense refers to potential delays. There are six conditions that can cause delay:

- . waiting on user input
- . waiting on system output
- . waiting for Screener response
- . waiting on polling text (user input or system output)
- . waiting on Screener assignment
- . waiting for RNPE connection

Each time the WES user's delay state changes, FSMGP updates the user's GPROF file. This file is read by EMP whenever the EM requests a user profile. See Figure 3-8 for a GPROF record.

Each record has the following fields:

PUSR The one-byte FSM user ID.

PTTY The UNIX terminal ID.

PSUB The WWMCCS subsystem in use at the time the status was recorded.

PSTAT The user's status. Values are 0 (waiting on user input), 1 (waiting on system output), 2 (waiting on Screener response), 3 (waiting on polling text), 4 (waiting on Screener assignment) and 5 (waiting on RNPE/Honeywell H6000 connection).

PDTG The four-byte DTG of the status.

Word 0	PUSR	PTTY
Word 1	PSUB	PSTAT
Word 2	PDTG	
Word 3		

Figure 3-8. A GPROF Record



3.1.1.2.1.11 Global Audit Description (GAUD) File. In order for ATDP to know quickly what audit files are present, in addition to the total time range they cover, a file exists which contains all audit file-pairs (one entry per pair), including the current pair, their DTG of creation and DTG of last entry. Each time FAD logs an audit event in the current file-pair, it must also update the corresponding GAUD record with the new "last entry" DTG. Since the description record (in GAUD) for the current file-pair is always the last record in the file, FAD has no problem updating it.

The description file contains fixed-length sequential records; one record per audit file-pair. Each record follows this format:

AUDFN Audit file-pair name (14 bytes).  
CRET Date-time group of creation (4 bytes).  
LMOD Date-time group of last entry (4 bytes).

In addition to FAD, the description file is also updated by ATDP. When ATDP saves the current file-pair, it modifies the description record of the current file-pair by changing the filename from "current" to the DTG of the save. It also writes a new record with a filename of "current" and the DTG of creation. Notice, ATDP does NOT modify the DTG of last entry on the old current file. This data is only modified by FAD. Similarly, FAD never modifies the filename or the DTG of creation. Since both FAD and ATDP have write-access to this file, it must be locked before writing. See Figure 3-9 for a GAUD record.

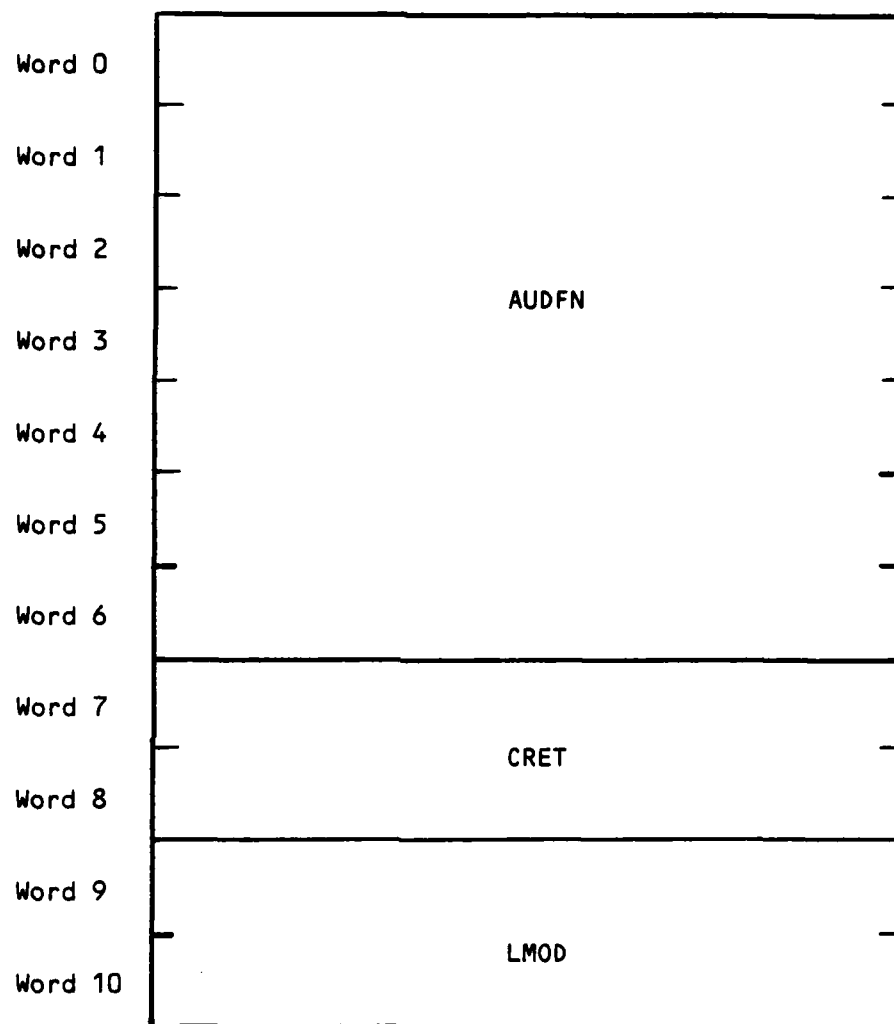


Figure 3-9. A GAUD Record

3.1.1.2.1.12 Audit File Lock (GAUD-LOCK). As mentioned previously, to protect audit files from being simultaneously write-accessed, FSM employs a standard lock-for-writing mechanism. The mechanism involves requiring all writing processes to lock the file before writing and unlocking it after. The existence of the lock file signifies a process is currently updating an audit file. The two file sets which are of concern are the Current Audit File-Pair and the GAUD file. Both are updated by ATDP and FAD. The lock file has no records, it is merely an entry in the audit directory so that its presence or absence can be detected easily.

3.1.1.2.1.13 Current Global Audit Event (GAEV) Files. The Global Audit Event Files are pairs of sequential files which contain records of varying format and length. All audit events are written to the GAEV files. These events can be generated by any one of the four FSM processes: FSM Guardian Process (FSMGP), Environment Manager Process (EMP), Screener Trusted Process (SCTP) or FSM Assignment Daemon (FASSD). Audit events are actually audited (written to the GAEV files) by the FSM Audit Daemon (FAD). User-oriented formatting and display of GAEV records is performed by the Audit Trail Display Program (ATDP).

There are nineteen events logged by FAD:

Event Code	Event Description
01	User input sent to system (WWMCCS)
02	User input not sent to system
03	System output matched and sent to user
04	System output not matched
05	System output accepted by Screener
06	System output rejected by Screener
07	Screener logged onto FSM
08	Screener logged off FSM
09	User logged onto FSM
10	User logged off FSM
11	User initiated a WWMCCS subsystem
12	User assigned to Screener
13	User input simulated (resynch)
14	Screen mode modified
15	Bandwidth threshold modified
16	EM logged onto FSM
17	EM logged off FSM
18	System output received from WWMCCS
19	Screener begins downgrade request

The GAEV file-pair consist of two files: the Event file and the Text file. Every audit event causes a record to be written to the Event file. Additionally, some audit events will write ASCII strings to the Text

file. All auditable user input and system output will be written to the Text file with the appropriate offset and length entered with the corresponding Event file record. For example, consider audit event 01 (User input sent to system). This event causes a record to be written in the Event file containing useful data (user ID, DTG and so on), as well as the offset in the Text file where the user input can be found and its length. The motivation behind two files, as opposed to one, stems from the potential size of user inputs and system outputs; as well as the fact that not all audit events have text to audit.

All of the following event records are written to the Event file. Also, the following descriptions pertain only to the Event file, NOT the Text file. Records in the Text file are format-less; that is, they are variable-length ASCII strings.

Each audit event record contains a fixed-size header block followed by a variable length Event Description. Figure 3-10 illustrates an audit event record.

The header is the same for all audit events and contains these fields:

FEVENT The Event Code (01-19).

FEVSIZ The number of bytes in the Event Description. NOTE: This does not include the size of the header.

FEVTIM The Date-Time Group (DTG) of this event in standard UNIX two integer (four bytes) format.

Figure 3-11 is an Event file Record Header Description.

HEADER
EVENT DESCRIPTION

Figure 3-10. An Audit Event Record (written to Event File)

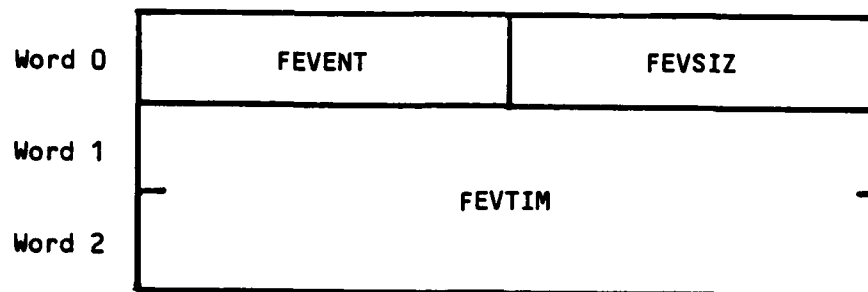


Figure 3-11. An Event File Record Header Description

The following paragraphs provide a description of the content and format of the audit event record for each of the nineteen events recorded in the GAEV files.

#### EVENT 01 (User input sent to system)

FSMGP will generate this audit event whenever the FSM has recognized and allowed user input to be sent to a WWMCCS subsystem. The user input will be copied to the Text file with the text file offset and text length entered in the Event file. Figure 3-12 is an Event 01 record description to be written to the Event file.

#### Field Descriptions

USERID The one byte FSM user ID.

USRTTY The UNIX identifier of the terminal from which the input was sent.

SUBSYS The current WWMCCS subsystem in use. See Table 3-2 for the

OFFSET The byte offset from the beginning of the Text file where the user input is written.

LENGTH The byte length of the user input.



Word 0	FEVENT	FEVSIZ
Word 1	FEVTIM	
Word 2		
Word 3	USERID	USRTTY
Word 4	SUBSYS	
Word 5	OFFSET	
Word 6		
Word 7	LENGTH	
Word 8		

Figure 3-12. Description of Event 01, 03, 04 , 13 and 18 Records

Table 3-2. WWMCCS Subsystems and their FSM codes

Subsystem Code	WWMCCS Subsystem
0	ALL subsystems
1	ACCESS
2	LIST
3	SIOS (Standard Input/Output System)
4	TCON (Transaction Constructor)
5	TLCF (Teleconferencing)
6	TSS (Timesharing System)
7	WWDMS (World-Wide Data Management System)
8	Logon

EVENT 02 (User input not sent to system)

Whenever FSMGP cannot recognize the user input or the user attempts to use a privileged WWMCCS facility that is not allowed, FSMGP will generate this audit event. The user input will be written to the Text file. Figure 3-13 is an Event 02 record description to be written to the Event file.

Field Descriptions

USERID The one byte FSM user ID.

Word 0	FEVENT	FEVSIZ
Word 1	FEVTIM	
Word 2		
Word 3	USERID	USRTTY
Word 4	SUBSYS	FAILEX
Word 5	OFFSET	
Word 6		
Word 7	LENGTH	
Word 8		

Figure 3-13. Description of Event 02 Record

USRTTY The UNIX identifier of the terminal from which the input was sent.

SUBSYS The current WWMCCS subsystem in use. Refer to Table 3-2 for code explanations.

FAILEX The coded reason why the input was not sent. Codes are: 1=No match (unrecognizable input) and 2=Match but disallowed.

OFFSET The byte offset from the beginning of the Text file where the user input can be found.

LENGTH The byte length of the user input.

#### EVENT 03 (System output matched and sent to user)

Whenever system output is matched (recognized) by the FSMGP and sent to the user, thereby bypassing the Screener, FSMGP generates this audit event. The system output is written to the Text file. Refer to Figure 3-12 for an Event 03 record description to be written to the Event file.

#### Field Descriptions

USERID The one byte FSM user ID.

USRTTY The UNIX identifier of the terminal to which the output was sent.

SUBSYS The current WWMCCS subsystem in use. Refer to Table 3-2 for code explanations.

OFFSET The byte offset from the beginning of the Text file where the system output was written.

LENGTH The byte length of the system output.

#### EVENT 04 (System output not matched)

In the course of a system-user dialogue, whenever the system responds with unrecognizable output, FSMGP will generate this audit event. The system output will be written to the Text file. Refer to Figure 3-12 for an Event 04 record description to be written to the Event file.

#### Field Descriptions

USERID The one byte FSM user ID.

USRTTY The UNIX identifier of the terminal to which the output would have been sent.

SUBSYS The current WWMCCS subsystem in use. Refer to Table 3-2 for code explanations.

OFFSET The byte offset from the beginning of the Text file where the system output was written.

LENGTH The byte length of the system output.

#### EVENT 05 (System output accepted by the Screener)

When a WWMCCS subsystem returns output which must be visually inspected by a Screener for downgrading and the output was accepted for downgrading, FSMGP generates this audit event. The accepted system output is written to the Text file. See Figure 3-14 for an Event 05 record description to be written to the Event file.

Word 0	FEVENT	FEVSIZ
Word 1	FEVTIM	
Word 2		
Word 3	USERID	USRTTY
Word 4	SUBSYS	SCRID
Word 5	OFFSET	
Word 6		
Word 7	LENGTH	
Word 8		

Figure 3-14. Description of an Event 05 and 06 Records

#### Field Descriptions

USERID The one byte FSM user ID.

USRTTY The UNIX identifier of the terminal to which the accepted output was sent.

SUBSYS The current WWMCCS subsystem in use. Refer to Table 3-2 for code explanations.

SCRID The one byte FSM user ID of the Screener.

OFFSET The byte offset from the beginning of the Text file where the system output is written.

LENGTH The byte length of the accepted system output.

#### EVENT 06 (System output rejected by Screener)

In this case, a WWMCCS subsystem output was visually inspected by a Screener and the downgrade request was refused. FSMGP will be notified of the downgrade rejection and will then generate this audit event. The rejected system output is written to the Text file. Refer to Figure 3-14 for an Event 06 record description to be written to the Event file.

#### Field Descriptions

USERID The one byte FSM user ID.

USRTTY The UNIX identifier of the terminal to which the output was to be sent.

SUBSYS The current WWMCCS subsystem in use. Refer to Table 3-2 for code explanations.

SCRID The one byte FSM user ID of the Screener.

OFFSET The byte offset from the beginning of the Text file where the rejected system output is written.

LENGTH The byte length of the rejected system output.

#### EVENT 07 (Screener logon to FSM)

Whenever a Screener logs onto the FSM (through a new instance of SCTP), SCTP will generate this audit event. See Figure 3-15 for an Event 07 record description to be written to the Event file.

##### Field Descriptions

SCRID The one byte FSM user ID of the Screener.

SCRTTY The UNIX identifier of the Screener terminal logging on.

#### EVENT 08 (Screener logoff of FSM)

When a Screener logs off of the FSM, FASSD will generate this audit event. Refer to Figure 3-15 for an Event 08 record description to be written to the Event file.

##### Field Descriptions

SCRID The one byte FSM user ID of the Screener.

SCRTTY The UNIX identifier of the Screener terminal logging off.



Word 0	FEVENT	FEVSIZ
Word 1	FEVTIM	
Word 2		
Word 3	SCRID	SCRTTY

Figure 3-15. Description of Event 07 and 08 Records

#### EVENT 09 (User logon to FSM)

When a WES user begins a terminal session (which includes logon sequence) FSMGP will generate this audit event. See Figure 3-16 for an Event 09 record description to be written to the Event file.

##### Field Descriptions

USERID The one byte FSM user ID.

USRTTY The UNIX identifier of the user terminal logging on.

NOTE: A user who fails the initial logon sequence may not have any audit events generated.

#### EVENT 10 (User logoff of FSM)

When a WES user ends a terminal session, FASSD will generate this audit event. This event involves two cases: explicit user logoff and user logoff due to error conditions. Refer to Figure 3-16 for an Event 10 record description to be written to the Event file.

##### Field Descriptions

USERID The one byte FSM user ID.

USRTTY The UNIX identifier of the terminal logging off.

Word 0	FEVENT	FEVSIZ
Word 1	FEVTIM	
Word 2		
Word 3	USERID	USRTTY

Figure 3-16. Description of Event 09 and 10 Records

#### EVENT 11 (User initiates a WWMCCS subsystem)

When a user initiates a WWMCCS subsystem, FSMGP will generate this audit event. A WWMCCS subsystem can be initiated in response to the "SYSTEM ?" prompt or at other less well-defined times. For instance, if the user is currently using the TCON subsystem, he can invoke the LIST subsystem at any time. NOTE: Logon is not considered a valid WWMCCS subsystem for this event. See Figure 3-17 for an Event 11 record description to be written to the Event file.

#### Field Descriptions

USERID The one byte FSM user ID.

USRTTY The UNIX identifier of the terminal initiating a WWMCCS subsystem.

SUBSYS The initiated subsystem code. Refer to Table 3-2 for code explanations.

STATUS The status of the initiation attempt (0=Failure, 1=Success).

Note: If STATUS=0, SUBSYS will be undefined.

Word 0	FEVENT	FEVSIZ
Word 1	FEVTIM	
Word 2		
Word 3	USERID	USRTTY
Word 4	SUBSYS	STATUS

Figure 3-17. Description of Event 11 Record

## EVENT 12 (User assigned to Screener)

Whenever the FASSD makes an initial WES user/Screener assignment or the EM (via the EMP) reassigns a WES user to a Screener, FASSD generates this audit event. See Figure 3-18 for an Event 12 record description to be written to the Event file.

### Field Descriptions

USERID The one byte FSM user ID of the assigned user.

USRTTY The UNIX identifier of the assigned user terminal.

SCRID The one byte FSM user ID of the assigned Screener.

SCRTTY The UNIX identifier of the assigned Screener terminal.

Word 0	FEVENT	FEVSIZ
Word 1	FEVTIM	
Word 2		
Word 3	USERID	USRTTY
Word 4	SCRID	SCRTTY

Figure 3-18. Description of Event 12 Record

#### EVENT 13 (User input simulated)

When the FSMGP receives system output that is unrecognizable, FSMGP will simulate user input to the WWMCCS subsystem in use in order for FSMGP to keep from getting lost in the context tables. This usually involves "popping up" to the next higher level within WWMCCS. This could terminate the current subsystem (thus, returning the user to the "SYSTEM?" prompt) or return to a well-defined point within the current subsystem (eg. the "NEXT?" prompt in TCON). The simulated user input is written to the Text file. Refer to Figure 3-12 for an Event 13 record description to be written to the Event file.

#### Field Descriptions

USERID The one byte FSM user ID.

USRTTY The UNIX identifier of the terminal from which input is being simulated.

SUBSYS The current WWMCCS subsystem in use. Refer to Table 3-2 for code explanations.

OFFSET The byte offset from the beginning of the Text file where the simulated user input is written.

LENGTH The byte length of the simulated user input.

#### EVENT 14 (Screen mode modified)

When the EM (via the EMP) modifies the downgrade screening mode, EMP generates this audit event; or if a WES user's FSMGP detects that the user's



bandwidth threshold has been exceeded, FSMGP modifies the user's screen mode (to ALL) and generates this event. Screen mode may be set to any one of three values:

NONE No output is inspected by the Screener.

NORMAL Only output FSMGP cannot match is inspected by the Screener.

ALL All output is inspected by the Screener.

Additionally, the screen mode may be set for a user, for a subsystem or for a user/subsystem. See Figure 3-19 for an Event 14 record description to be written to the Event file.

#### Field Descriptions

USERID The one byte FSM user ID. If ALL users are affected, this value will be set to -1.

USRTTY The UNIX identifier of the user's terminal. This value is undefined if USERID is -1.

SUBSYS The WWMCCS subsystem(s) for which the screen mode was modified. Refer to Table 3-2 for code explanations. The logon subsystem is not affected by the screening mode.

SCMODE The new screen mode (0=NONE, 1=NORMAL and 2=ALL).

Word 0	FEVENT	FEVSIZ
Word 1	FEVTIM	
Word 2		
Word 3	USERID	USRTTY
Word 4	SUBSYS	SCMODE

Figure 3-19. Description of Event 14 Record

#### EVENT 15 (Bandwidth Threshold modified)

When the EM (via the EMP) modifies the bandwidth threshold for a user, EMP generates this audit event. The bandwidth threshold is defined as the number of messages allowed to be automatically downgraded for a user each minute. If the bandwidth threshold is surpassed, screen mode is automatically modified to ALL for the particular user (by FSMGP). See Figure 3-20 for an Event 15 record to be written to the Event file.

##### Field Descriptions

**USERID** The one byte FSM user ID. If ALL users are affected, this value will be -1.

**RATE** The number of messages per minute which may automatically bypass the Screener. The rate must be a positive integer in the range of one to 127.

#### EVENT 16 (EM logon to FSM)

When the EM logs onto FSM, EMP generates this audit event. See Figure 3-21 for an Event 16 record to be written to the Event file.

##### Field Descriptions

**EMID** The one byte FSM user ID of the Environment Manager.

**EMTTY** The UNIX identifier of the EM's terminal.

Word 0	FEVENT	FEVSIZ
Word 1	FEVTIM	
Word 2		
Word 3	USERID	RATE

Figure 3-20. Description of Event 15 Record

#### EVENT 17 (EM Logoff of FSM)

When the EM logs off of FSM, EMP generates this audit event. See Figure 3-21 for an Event 17 record to be written to the Event file.

##### Field Descriptions

EMID The one byte FSM user ID of the Environment Manager.

EMTTY The UNIX identifier of the EM's terminal.

#### EVENT 18 (System output received by FSMGP)

When FSMGP receives a system output from WWMCCS, via the RNPE, and has matched a halting delimiter (the output may or may not subsequently match a SUBCONTEXT), this event is generated. The system output received is written to the Text file. Refer to Figure 3-12 for an Event 18 record description.

##### Field Descriptions

USERID The one byte FSM user ID of the user.

USRTTY The UNIX identifier of the user's terminal.

SUBSYS The current WWMCCS subsystem in use. Refer to Table 3-2 for code explanations.

OFFSET The byte offset from the beginning of the Text file where the system output can be found.

LENGTH The byte length of the user input.

Word 0	FEVENT	FEVSIZ
Word 1	FEVTIM	
Word 2		
Word 3	EMID	EMTTY

Figure 3-21. Description of Event 16 and 17 Records

AD-A171 332

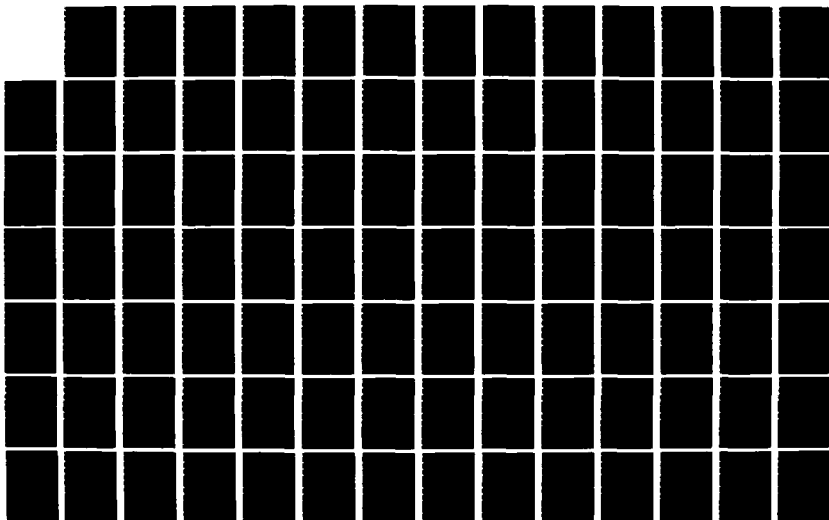
FORSYTH SECURITY MONITOR (FSM) COMPUTER PROGRAM  
DEVELOPMENT SPECIFICATION (TYPE B5)(U) LOGICOM INC SAN  
DIEGO CA FEB 81 N88039-78-C-0323

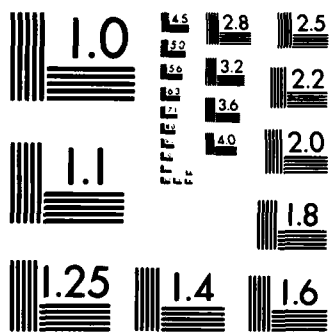
2/4

UNCLASSIFIED

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A



#### EVENT 19 (Downgrade request received by Screener)

When the Screener begins reviewing a downgrade request from a WES user this event is generated by SCTP. The time of this event in conjunction with the time of the subsequent event 05 (accept) or 06 (reject) yields the time needed by the Screener to make his decision. See Figure 3-22 for an Event 19 record description.

#### Field Descriptions

SCRID The one byte FSM user ID of the Screener.

SCRTTY The UNIX identifier of the Screener's terminal.

SUBSYS The current WWMCCS subsystem in use. Refer to Table 3-2 for code explanations.

USERID The one byte FSM user ID of the WES user requesting the downgrade.

3.1.1.2.1.14 Current Audit Text File. When audit events are logged which contain audit text (system output or user input), the text is written in the current audit text file. It has no format. Text is written to the file, by being appended, as ASCII strings. The starting location (OFFSET) and LENGTH of each event's text is recorded with the corresponding event record in the current GAEV file.

Word 0	FEVENT	FEVSIZ
Word 1	FEVTIM	
Word 2		
Word 3	SCRID	SCRTTY
Word 4	SUBSYS	USERID

Figure 3-22. Description of Event 19 Record

3.1.1.2.1.15 Saved Global Audit Environment (GAEV) Files. ATDP can save the current audit file-pair and initialize a new pair. Once a file-pair has been saved, it can be accessed only by ATDP and only for reading. Therefore, saved audit files (i.e. non-current) need no corresponding lock files. A saved audit file-pair is renamed to the DTG of the save operation. For example, saving a current file-pair on April 2 at 12:05:30 has the effect of renaming the current Event file from "current.e" to "0402120530.e" and the current Text file from "current.t" to "0402120530.t". The new current file-pair exists with the same name as before but now with zero length. NOTE: Do not confuse the saved DTG with the DTG of the last-entered audit event. Records in the saved Event files have formats as described above in the Global Audit Event (GAEV) file description. Records in the saved Text files are format-less.

3.1.1.2.1.16 Saved Audit Text Files. Saved audit text files have no format. They contain ASCII strings which were audit text. Associated with each saved text file is a saved GAEV file.

3.1.1.2.1.17 Audit Request Files. After an audit event has been generated by one of the four primary FSM processes, GFAUDIT is called by the generating process and passed pertinent event data items. It processes its arguments and writes a file containing an audit request into the audit queue. The audit request file contains all known event data that is needed by FAD to log the audit event into the current audit files. The file has a name composed of the Date-Time Group (DTG) of the audit event, the process ID of the event generator (one of FSMGP, EMP, SCTP or

FASSD) and a random alphanumeric character. This format guarantees unique filenames in the queue. Specifically, the filename format is:

MMDDhhmmsspppr

where,

MM is the month (01-12)

DD is the day (01-31)

hh is the hour (00-23)

mm is the minute (00-59)

ss is the second (00-59)

ppp is the process ID of the event generator (000-999)

r is a random alphanumeric character

The audit request file written by GFAUDIT is composed of one or more records. The first record of every file will follow the format of the audit event it represents. These follow the formats described above in the Global Audit Event (GAEV) file description. If the particular event also contains text, in the form of system output or user input, there will be a second and possibly additional records which contain the text. The text has no format; that is, it is composed of ASCII strings and is guaranteed to be terminated by a UNIX null byte ('0').

All audit request files are written into an audit request queue by the global audit function GFAUDIT. Audit request files are processed (the events are logged) by the audit daemon FAD. The audit request queue is implemented as a directory entry in the main audit directory "/fsm/fsat".

The queue pathname is `"/fsm/fsat/auditq"`.

An empty audit queue directory (i.e. no files) signifies that there are no pending audit requests. Similarly, `n` files signifies `n` pending audit requests. Audit requests are handled in a strict First-in, First-out (FIFO) basis and are unlinked by FAD as the specified event is recorded in the current audit files.

3.1.1.2.1.18 Lock-File sclog-i. This lock-file provides a means for SCTP to inform assigned FSMGPs that a Screener is logging off or has logged off. This prevents downgrade requests from being sent to logged-off Screeners. Prior to requesting a downgrade, FSMGP checks for the presence of this lock-file. If the lock-file is not present, the downgrade request is sent to the assigned Screener. If the port exists, the downgrade request is deferred until a new Screener is assigned. In the latter case, the user is also notified that no Screener is available. The lock-file name consists of the characters "sclog" followed by the Screener's FSM user ID as a five character string with leading zeroes.

3.1.1.2.2 Port Descriptions. The inter-process communication (IPC) ports used by FSM are for the use of FSMGP, FASSD, EMP and SCTP. Ports are used to pass IPC messages indicating various events applicable to FSM, such as Screener assignments, Screener logon/logoff notifications, environment changes, and so on. Following are descriptions of the individual ports implemented in FSM and corresponding message formats.

**3.1.1.2.2.1 Port Message Descriptions.** The FSM port messages will consist of structured data, both binary and ASCII. See Table 3-3 for a summary of the message types. Figure 3-23 describes the actual structure of the message.

#### **Field Descriptions**

**GMTYP** One byte Message Type.

**GUID** One byte FSM user ID.

**GEENG** A one byte environmental change value (bypass rate or screen mode); or a second FSM user ID.

**GSSI** One byte subsystem index (refer to Table 3-2 for code explanations).

**GCLAS** Integer security classification code.

**GDPTH** Pathname of a downgrade file.

**GCPTH** Pathname of a context file.

Word 0	GMTYP	GUID
Word 1	GECNG	GSSI
Word 2	GCLAS	
Word 3	GDPTH	
	• • •	
Word 15		
Word 16	GCPTH	
	• • •	
Word 28		

Figure 3-23. Description of IPC Messages

Table 3-3. Summary of IPC Messages

General Format:

<mtyp>[<uid>][<ecng>[<ssinx>][<sc><dgfn><ctxfn>]

where:

- \* Items enclosed in [] are optional.
- \* "mtyp" represents the IPC message type (byte).
- \* "uid" represents a user ID (byte).
- \* "ecng" represents an environment change, a bypass rate change (number of messages per minute), a screen mode change (0 - none, 1 - normal, 2 - all); or a second user ID (all values one byte).
- \* "ssinx" represents a subsystem index for a screen mode change (byte).

0 - All subsystems

1 - ACCESS

2 - LIST

3 - SIOS

4 - TLCF



5 - TLCF

6 - TSS

7 - WWDMS

8 - logon

\* "sc" represents the security classification of the user (byte).

1 - unclassified

2 - confidential

3 - official use only

4 - secret working papers

5 - secret

\* "dgfn" represents the pathname of a file to be downgraded.

\* "ctxfn" represents the pathname of a context file associated with a downgrade file.

IPC Message Types are:

UAREQ <uid><ecng>                      User assignment request.

UALGO <uid>                              User logoff notification.

SCASS <uid><ecng>                      Screener assigned.

SCNAV	Screeners not available.
SCCNG <uid><ecng>	Screeners assignment change.
BRCNG <ecng>	Bypass rate change.
SMCNG <ecng><ssinx>	Screen mode change.
DGREQ <uid><ecng><sc><dgfn><ctxfn>	Downgrade request.
DGACC <uid>	Downgrade accepted.
DGREJ <uid>	Downgrade rejected.
DRRDY <uid><ecng>	Screeners ready (logon).
DRLGO <uid>	Screeners logoff notification.

3.1.1.2.2 Port faport. This port provides a means for a FSMGP to request a Screeners assignment for a newly logged on WES user. FSMGP writes a UAREQ message to FASSD. FASSD writes a reply (SCASS or SCNAV) on the appropriate fsagn-i port. Sctp also uses the faport to notify FASSD of Screeners logons/logoffs (DRRDY/DRLGO). Finally, EMP utilizes this port to notify FASSD of Screeners reassignments (SCCNG).

3.1.1.2.2.3 Port fsagn-i. This port provides a means for FASSD and EMP to communicate with an FSMGP. The port name consists of the characters "fsagn" followed by the WES user's FSM ID as a five character string with leading zeros. FASSD writes messages SCASS and SCNAV on this port to one or more of the currently running FSMGPs. Messages SCASS and SCNAV are

replies to the request message UAREQ which FASSD reads from its input port faport. Messages BRCNG and SMCNG can be written at any time on this port by the EMP.

3.1.1.2.2.4 Port screq-i. This port provides a means for a FSMGP to request a downgrade (DGREQ). The port name consists of the characters "screq" followed by the Screener's user ID as a five character string with leading zeroes. The FSMGP writes a DGREQ message to the appropriate SCTP using this port. The SCTP will write a reply message (DGACC, DGREJ or DRLGO) on the appropriate fsrly-i port.

3.1.1.2.2.5 Port fsrly-i. This port provides a means for a SCTP to communicate with a FSMGP. The port name consists of the characters "fsrly" followed by the WES user's FSM ID as a five character string with leading zeroes. The SCTP writes messages DGACC, DGREJ and DRLGO on this port to the appropriate FSMGP. These are reply messages to a DGREQ message which the SCTP read from its input port screq-i.

### 3.1.2 Process/Data Structure Interfaces

To this point we have described FSM's data structures in great detail but have only given a simplistic view of how they are manipulated by FSM processes. In this subsection we now describe how and when processes access FSM data structures and/or communicate using them (from a data structure viewpoint). For particulars of data access from the process view, see Section 3.2. The descriptions presented below refer to Figures 3-24 and 3-25, disk files and ports, respectively.

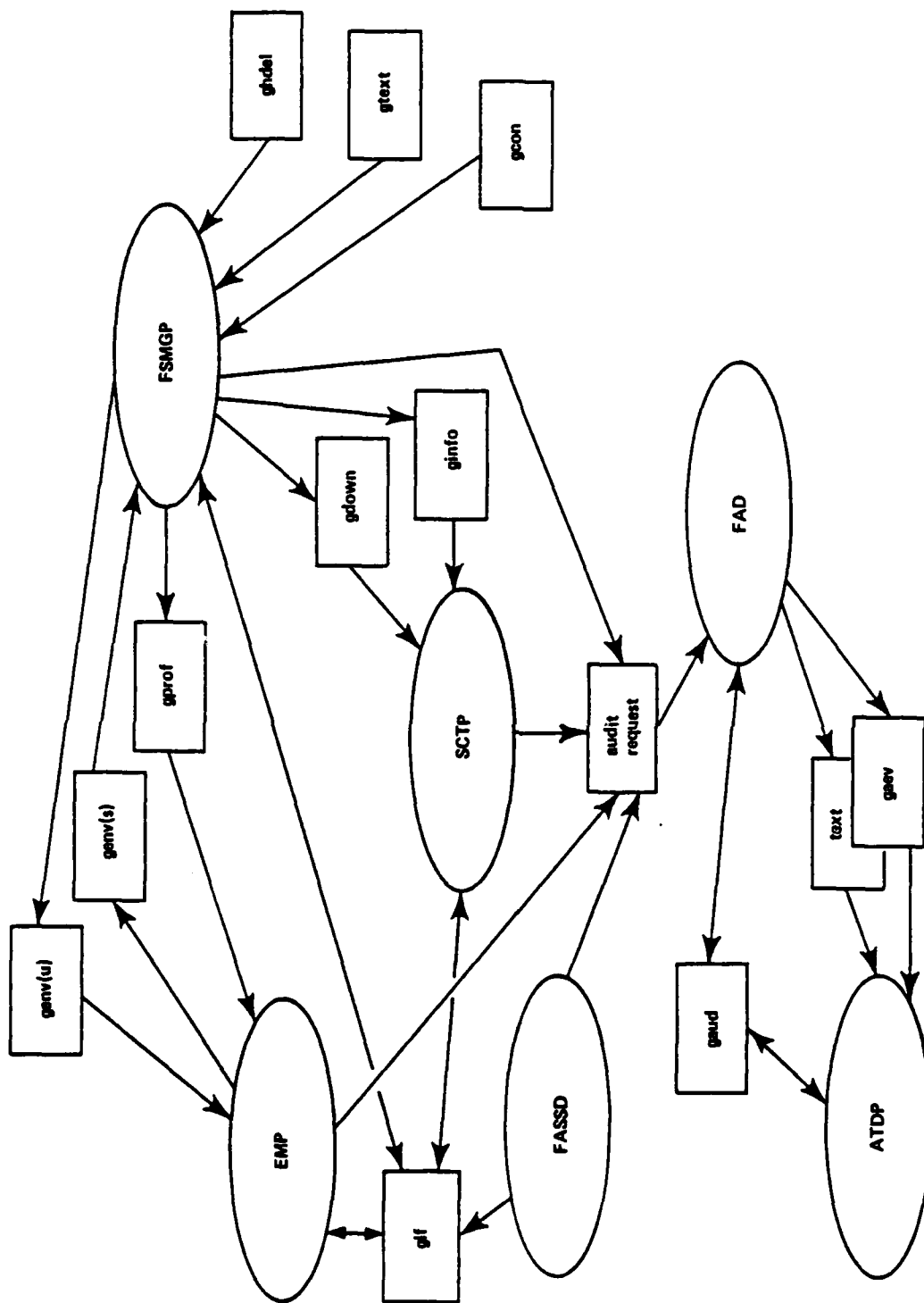


Figure 3-24. FSM Process/Disk File Flow

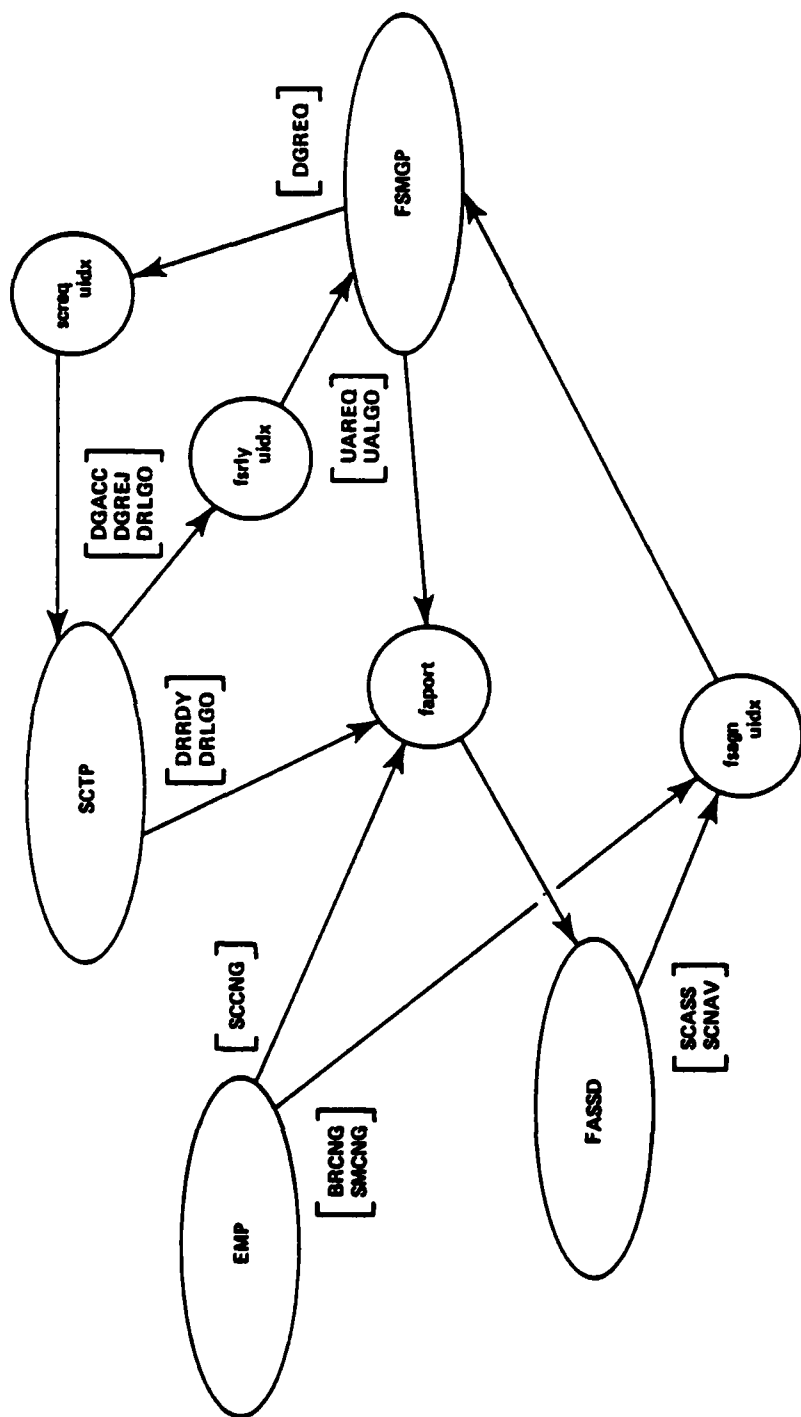


Figure 3-25. FSM Process/Port Flow

3.1.2.1 Disk File Interfaces. All FSM disk file data structures will be described with respect to process manipulation. These disk files are:

- a. Context Data (GCONT)
- b. Context Text Data (GTEXT)
- c. Context Halting Delimiter Data (GHDEL)
- d. User Profile Data (GPROF)
- e. System Environment Data (GENV)
- f. User Environment Data (GENV)
- g. User Logon File (GLF)
- h. Downgrade File (GDOWN)
- i. Downgrade Context Information File (GINFO)
- j. Audit Request Data
- k. Audit File (GAUD)
- l. Audit Event Data (GAEV)
- m. Audit Text Data
- n. Screener Lock-File (sclog-i)

3.1.2.1.1 Context Data (GCONT). FSM Context Data (GCONT) is manipulated by only one FSM process type, the FSM Guardian Program (FSMGP). This data structure contains the entire user/WWMCCS dialogue context and controls all automatic/manual screening and filtering decisions. It is read-only and is used to control the referencing of its related data structures, the Context Text Data (GTEXT) and the Context Halting Delimiter Data (GHDEL). Since this data structure is read-only it does not need to be locked during FSM operation.

3.1.2.1.2 Context Text Data (GTEXT). FSM Context Text Data (GTEXT) is manipulated by only one FSM process type, FSMGP. This data structure contains the textual data of the user/WWMCCS dialogue context and is pointed to by appropriate entries in GCONT. Since this data structure is read-only it does not require locking sequences.

3.1.2.1.3 Context Halting Delimiter Data (GHDEL). FSM Context Halting Delimiter Data (GHDEL) is referenced by only one FSM process type, FSMGP. This data structure contains halting delimiter strings utilized by FSMGP in recognizing user/system data sequences. As with its related data structures GCONT and GTEXT, this data structure does not require locking sequences.

3.1.2.1.4 User Profile Data (GPROF). The User Profile Data (GPROF) structure is created by FSMGP for each WES user logged onto FSM. GPROF contains the current state the WES user is in during his dialogue with WWMCCS. GPROF is updated by FSMGP each time the WES user enters another profile state. The Environment Manager Program (EMP) reads GPROF when

the Environment Manager (EM) wishes to view the profile of some WES user logged onto FSM.

3.1.2.1.5 System Environment Data (GENV). The Environment Data (system) (GENV) structure is created when the FSM is initiated and contains a set of default environment values. The Environment Manager Program (EMP), under direction of the EM, updates this data structure to reflect new system-wide environment values. EMP also displays GENV to the EM. FSMGP utilizes the system GENV to establish initial local environment values for its local user (it is used to build the user level GENV).

3.1.2.1.6 User Environment Data (GENV). The Environment Data (user) Structure is created by FSMGP when a WES user logs onto FSM. FSMGP uses the system GENV data structure to initialize the user's local GENV. As FSMGP receives new environment values from EMP, it updates the user's GENV accordingly.

3.1.2.1.7 Global Logon File (GLF). The Global Logon File (GLF) exists when FSM is initiated and is updated by the FSM Guardian Process (FSMGP), FSM Assignment Daemon (FASD), Screener Trusted Process (SCTP), Environment Manager Process (EMP) and a stand-alone maintenance program, USER (see Appendix E). When a WES user logs onto FSM, FSMGP updates GLF to reflect that fact. SCTP operates similarly when a Screener logs on. FASD modifies GLF when WES users or Screeners log off. When an EM logs on or off, EMP modifies GLF. Additionally, FASD modifies GLF when an assignment is made and when an EM-directed reassignment occurs. USER modifies GLF whenever it is directed to do so.



3.1.2.1.8 Downgrade File (GDOWN). The Downgrade File (GDOWN) is created by the FSM Guardian Program (FSMGP) when a user logs onto FSM. This data structure contains WWMCCS output for manual screening by the FSM Screener assigned to the WES user. Therefore the Screener Trusted Process (SCTP) must read GDOWN.

3.1.2.1.9 Downgrade Context File (GINFO). The Downgrade Context File (GINFO) is created by FSMGP when a WES user logs onto FSM. This data structure contains dialogue context data used by the FSM Screener in making a decision as to the sensitivity of the downgrade data (i.e., GDOWN). Under Screener control, SCTP reads GINFO, displaying a screen at a time to the Screener.

3.1.2.1.10 Audit Request Data. Audit Request Data is created as separate event/text file entries by four FSM processes: the FSM Guardian Program (FSMGP), the Environment Manager Program (EMP), the Screener Trusted Program (SCTP) and the FSM Assignment Daemon (FASSD). FSMGP builds event/text entries for many different audit events, such as entering new WWMCCS subsystems, inputting commands, and so on. EMP creates audit event entries when the EM alters the environment. SCTP audits Screener logon and whenever a downgrade request is accepted for inspection. FASSD creates audit events when a WES user or Screener logs off of FSM or a WES user/Screener assignment is made. The FSM Audit Daemon (FAD) is responsible for updating the current FSM Audit Trail with these audit request data. FAD deletes the request data after it is recorded in the current audit trail.

3.1.2.1.11 Audit File (GAUD). The Audit File (GAUD) exists when FSM is initiated and is updated by the Audit Trail Display Program (ATDP) under direction of the EM. GAUD reflects the current and old (saved) audit trail events/texts. When the EM desires to save the current audit trail, EMP updates GAUD to reflect the saved event/text pair. FAD also updates GAUD to reflect information concerning the current audit trail event/text pair. Since both ATDP and FAD update GAUD, it requires a lock sequence.

3.1.2.1.12 Audit Trail Event Data (GAEV) Structure. The Audit Trail Event Data (GAEV) Structure is created by the FSM Audit Daemon (FAD) and contains current audit trail events. EMP, under direction of the EM, can display this data in concert with its text file counterpart. When the EM wishes to save the current audit trail event/text pair, EMP creates another copy of these data structures and recreates the current GAEV to zero length. Although both ATDP and FAD have write access to the current GAEV, a lock sequence prevents an update conflict.

3.1.2.1.13 Audit Trail Text Data Structure. The Audit Trail Text Data Structure is handled identically as GAEV. It is created by FAD and is displayed by ATDP. ATDP also deletes the current version of this data structure when the EM commands it to be saved.

3.1.2.1.14 Screener Lock-File (sclog-i). The sclog-i lock-file is created when an active Screener is logging off. Its presence indicates that the owner (Screener) is unavailable for any further downgrade requests. Prior to sending his assigned Screener a downgrade request IPC message, FSMGP checks for the existence of this port (the "-i" is the

Screeners' FSM ID and uniquely identifies the file). If it is present, FSMGP must defer its downgrade request until another Screener is assigned. Otherwise, the downgrade request can be made as usual.

**3.1.2.2 Port Interfaces.** Figure 3-25 illustrates the FSM port/process interface. Ports are used in FSM as FIFO message queues and provide an efficient mechanism for inter-process communication (IPC). FSM ports are:

- a. FSM Assignment Daemon (faport) Port
- b. Screener Assignment (fsagn-i) Port
- c. Screener Request (screq-i) Port
- d. Screener Reply (fsrly-i) Port

Following is a description of these ports with respect to their utilization by FSM processes.

**3.1.2.2.1 FSM Assignment Daemon (faport) Port.** The faport port is created by the FSM Assignment Daemon (FASDD) and is utilized by existing FSM Guardian Processes (FSMGP) for requesting WES user/Screener assignments, Screener Trusted Processes (SCTP) for Screener logons/logoffs, and Environment Manager Process (EMP) for reassignment requests.

3.1.2.2.2 Screener Assignment (fsagn-i) Port. The fsagn-i port is created by FSMGP when it is invoked by a WES user. This port is written to by FASSD to pass WES user/Screener assignments and by EMP to indicate new environment values for the WES user. This port type exists for each WES user logged onto FSM and therefore utilizes the user's identifier in its name (i.e., "-i" means the "ith" WES user user ID). Port fsagn-i is deleted from FSM when the WES user logs off of FSM and FSMGP exits.

3.1.2.2.3 Screener Request (screq-i) Port. The screq-i port is created by the Screener Trusted Process (SCTP) and is written to by an assigned FSMGP for Screener requests. Since a FSM Screener can service more than one WES user, potentially many FSMGPs will be writing on any individual screq-i port. Note here, since more than one Screener can be logged onto FSM, the Screener's user ID is used in the port name (i.e., "-i" represents the "ith" Screener user ID). This port is deleted from the system when the Screener logs off of FSM and SCTP exits.

3.1.2.2.4 Screener Reply (fsrly-i) Port. The fsrly-i port is created by FSMGP and is written to by an assigned SCTP. In this situation, since a WES user can only have one Screener assigned to him at any one time, only one SCTP will be writing on this port. SCTP uses this port to inform the FSMGP of the Screener's downgrade decision (i.e., accept/reject) or of Screener logoff. This port is deleted from the system when the WES user logs off of FSM and FSMGP exits.

### 3.2 DETAILED FUNCTIONAL REQUIREMENTS

This section provides descriptions of the programs comprising the FORSCOM Security Monitor (FSM). Each program description is organized as follows. A short general description is followed by a list of the FSM global data used. Then, if applicable, a list of pertinent local data contained in the program is provided. A general process flow details the major paths the program traverses in accomplishing its task(s), with references to global and local data as well as major functions. The descriptions of major functions, if any, used in the general process flow discussion complete the program description.

The FSM programs described include:

- a. FORSCOM Security Monitor Guardian Program (FSMGP)
- b. Screener Trusted Program (SCTP)
- c. FSM Assignment Daemon (FASD)
- d. FSM Audit Daemon (FAD)
- e. Environment Manager Program (EMP)

f. Audit Trail Display Program (ATDP)

Notice, the Remote Network Processor Emulator (RNPE) is not described since it is furnished by a subcontractor and its internal operations are unknown.

3.2.1 FORSCOM Security Monitor Guardian Program (FSMGP)

The FORSCOM Security Monitor Guardian Program (FSMGP) is responsible for following user/system interactions in the restricted WWMCCS environment. It must know exactly where a user is in any given subsystem so that down-grade decisions can be made. Any given user input can be identified only in the context of what system output preceded it and what is to follow it. Conversely, any given system output can be understood only in terms of what has happened before and what the possible future events are. To do this tracking FSMGP must rely on a disk file, the context file, which contains all user inputs and system outputs, linked together in a tree-like structure. FSMGP is thus table driven, and its processing depends heavily on the information contained in the context file.

3.2.1.1 Global Data Usage. FSMGP uses the following global data files:

GCONT Context items and associated subcontexts.

GHDEL Context halting delimiter strings.

GTEXT Context text strings.

GENV System environment values.

GINFO Downgrade context information buffer.

GLF     Global logon file.

GPROF   Current processing state buffer.

3.2.1.2 Local Data Usage.   FSMGP uses the following pertinent local data:

IB       Working input buffer.

3.2.1.3 General Process Flow.   The FORSCOM Security Monitor Guardian Process (FSMGP) is activated when its parent process, FSMGTTY, receives user input from the terminal keyboard. FSMGP follows the standard WWMCCS TSS and TLCF logon procedures in the LOGON function which also establishes initial conditions. FSMGP receives the terminal identification as an argument and following local validation of the low-side WES user's WWMCCS userid, communicates to the FSM Assignment Daemon (FASDD), via an Inter-Process Communication (IPC) message on the port faport, requesting that a Screener be assigned to this user. FASDD communicates to FSMGP via the port fsagn, which is read only by the FSMGP for this user (the portname has a user identifier appended to it making it unique for this FSMGP). If there is no Screener available, the user is allowed to proceed until one is needed. Otherwise the Screener identification, user identification and terminal identification are written to the Global Logon File (GLF), using a lock mechanism. FSMGP then reads the system default Environment Values file (GENV) to determine the current default screen mode and bandwidth. This information is then written to the user's copy of the GENV, created by LOGON, so that FSMGP can update these values if the Environment Manager decides to change them. An audit event

is then passed to the Audit Function, GFAUDIT, indicating that a user has logged onto FSM. LOGON also creates the user Profile file (GPROF), Downgrade file (GDOWN) and the Downgrade Context file (GINFO).

LOGON processing then continues as if for a normal logon to WWMCCS TSS or TLCF systems. Upon successful completion of the logon sequence an event is passed to GFAUDIT recording that fact.

FSMGP then enters the main loop of its processing which continues until the user logs off. Since the loop is iterated each time a user input or a system output occurs, a new context must be established with each iteration. This is accomplished by reading the context file GCONT at the point established by the last context. The current context, with header and a variable amount of subcontexts, is copied into a buffer local to FSMGP so that quick access can be made to the information. Processing then takes alternate paths depending on whether the current context is system output or user input.

If it is a user context a loop is entered which reads inputs from the user until a match is made to one of the subcontext strings in the current context. Before FSMGP reads user input the GPROF file is updated to reflect that fact. If no match is found, an error message

\*\*\* UNRECOGNIZABLE OR DISALLOWED INPUT -- PLEASE TRY AGAIN \*\*\*

is issued to the user, the previous system output is re-written to the terminal, an audit event is recorded indicating invalid user input and the user is expected to enter new input. If a match is made the particular subcontext is checked for the presence of a special function. Special



functions for user subcontexts perform command syntax verification. If a special function is present it is executed. If the user's syntax was incorrect, it is treated the same as if it was invalid input (see above). If no error is detected the loop is exited and the user input is moved to the GINFO buffer, a local buffer to FSMGP that is continually written in a circular fashion enabling the most current user/system interactions to be saved. FSMGP then writes the matched user input to the system, audits that fact and establishes a new context in preparation for the system response.

If the current context is a set of expected system outputs, the System Output Match (SOM) function is called. This function updates GPROF to reflect the fact that FSMGP is waiting for system output, and reads system output from the Remote Network Processor Emulator (RNPE), comparing it to the set of subcontexts in the current context. If a match is found, the system output is written to the user's Downgrade file (GDOWN), which eventually is written to the user (if the system output is fixed), or the Screener (if the system output is variable) for downgrading. System output may be repeating or nonrepeating, which simply serves to indicate the size of a particular expected system output. Contexts of the repeating type are expected to be large, as in the case of TCON skeletons or TLCF messages, and oftentimes there is only one subcontext item. If the context indicates that the expected system output is repeating, SOM may make repeated reads to acquire the entire system output text. If the current context is variable repeating (e.g. TCON skeleton with data or a TLCF message), SOM accumulates system output line-by-line. If the

current context is fixed repeating (e.g. TCON skeleton without data), SOM acquires the entire system output matching each character until the terminating delimiter sequence is encountered.

If the current context is nonrepeating, system outputs are expected to be smaller and there may be more than one subcontext item. In this case SOM loops for the number of subcontexts or until a match is found to the current system output. If no match is found to the system output, SOM returns that fact.

If SOM returns with no match to a system output FSMGP audits the no match condition. Then a special function is executed which sends a simulated user input to the system to get FSMGP back in synchronization. This sending of simulated user input is known as a "pop-up". The special function used is indexed by the current context and performs a specific action depending on the particular place FSMGP is in the current subsystem. The current context also contains a pointer to the next context in the event that SOM returns no match. This resynchronization event is audited and FSMGP continues execution at the next context.

If SOM finds a match to the system output FSMGP must then decide whether or not it is to be sent to the Screener. System output may or may not be automatically screened by FSMGP. This depends on the current screen mode and bandwidth. FSMGP determines these values by first checking the port fsagn to see if the EM has changed either of these values. If so, the user's Environmental Values file (GENV) is updated. FSMGP switches on the screen mode value.

If the current screen mode is for no messages to be sent to the Screener, the system output contained in the GDOWN file is sent to the user's terminal. FSMGP then audits that fact, establishes a new context, and repeats its main execution loop.

If the current screen mode is for all messages to be sent to the Screener, the user's GPROF is updated to show that FSMGP is waiting for the Screener's decision, and a message is sent to SCTP via its port screq, requesting a downgrade. This message contains the name of the user's GDOWN, as well as the name of the user's GINFO. FSMGP then waits for the Screener's reply on the fsrly port, which contains the accept/reject decision. If the Screener accepts the system output, an audit event of that fact is recorded, the system output is sent to the user, a new context is established and FSMGP continues at the top of its main processing loop. If the Screener rejects the system output, an audit event is recorded, an error message is sent to the user, and the special function that resynchronizes FSMGP is executed.

If the current screen mode is for normal and the matched subcontext indicates that the current system output is fixed, the bandwidth rate is checked. This is accomplished in the CBR function, which computes a rate for the number of automatically screened messages that have occurred in a specific amount of time. If this bandwidth has been exceeded, the user's screen mode is automatically changed to "all", and his GENV is updated to reflect that fact. The system output is then sent to the Screener, and processing continues as above. If the bandwidth rate was not exceeded,

the system output is written to the user, an audit event is recorded, and processing continues in the main loop with a new context. Finally, if the current system output is variable, it is sent to the Screener and processing continues as above.

When the user terminates his session with WWMCCS, the LOGOFF function is called to terminate the program's execution. This function is invoked as a special function when the user enters a terminating sequence (e.g. bye) at an appropriate point. LOGOFF sends a logoff notification to FASSD, via the faport, so that FASSD may change GLF accordingly.

#### 3.2.1.4 Major Functions.

##### LOGON

The LOGON function is responsible for establishing FSMGP initial conditions and for tracking a user's progress through the WWMCCS TLCF and TSS logon procedures. LOGON writes an IPC to the FASSD, via the port faport, requesting that a Screener be assigned to the FSMGP's user. LOGON receives FASSD's reply on the port fsagn, and allows the user to continue until a Screener is necessary. (At that time, if there is no Screener available, the user is given the option of waiting for a Screener assignment (and subsequent downgrade decision) or logging off.) If a Screener has been assigned, this information, as well as the user's terminal identification, is written to GLF -- a Global Logon File with records for each valid user. Since this file is written to by every FSMGP during a user logon, a lock mechanism must be used. LOGON then reads the system default Environment Values file (GENV) to determine the

system-wide current screen mode and bandwidth rate. This information is then written to the user's copy of GENV. The user's GENV is maintained by FSMGP and is updated when it is discovered that the Environment Manager has changed one of these values. LOGON then calls the the global function GFAUDIT so that an audit event indicating that the user has logged on to WWMCCS can be recorded in the audit file. LOGON then processes a user's logon to WWMCCS TLCF or TSS systems in the normal fashion with two exceptions. The user's response to the "IDENT?" question is used to look-up this user's User Master Catalog (UMC) designation. All subsequent references to user-owned files, regardless of the WWMCCS subsystem involved, must be preceded by this UMC. Also, the response "zzz" to the system query "CLASSIFICATION OF YOUR OUTPUT?" is substituted for the user's response so that the paging and classification headers are not displayed by the system. LOGON also creates the user's copy of the Environment Values file (GENV), the user's Profile file (GPROF), the Downgrade file that is sent to the Screener (GDOWN), and the Downgrade Context file that the Screener may peruse during a downgrade (GINFO).

#### UIM

The User Input Match function (UIM) searches the subcontexts of the current context for a match to the user input that it reads upon its invocation. UIM loops for the number of subcontexts. If a match is found to the user's input the loop is exited. If a null subcontext is encountered it is saved so that it may be returned as a default match. Only one null subcontext per context set is allowed. A matched user input is moved to the Downgrade Context Buffer (GINFO), as well as

returned to FSMGP so that it can be written to the system. If no match is found, this fact is returned to FSMGP. FSMGP then issues an error message to the user and calls UIM again.

#### WTP

The Write to Profile function, WTP, updates the user's GPROF whenever FSMGP is waiting for the next user input, system output or Screener action. Additionally, during logon only, WTP is called to record the fact that FSMGP is waiting for an RNPE connection. Under normal circumstances, this event will occur once for a very short duration. WTP is called when a user input match search is made (by UIM), when a system output match search is made (by SOM), or when a downgrade request is made to the Screener (by STS). GPROF is created for the user by LOGON upon invocation of FSMGP. WTP opens GPROF, writes the index of the appropriate event (passed to it by FSMGP) as well as the current subsystem index (also passed to it by FSMGP) and current DTG. WTP then closes GPROF and exits.

#### WCB

The Write Context Buffer function, WCB, maintains the Downgrade Context Buffer (GINFO) in preparation for its eventual transference to the GINFO file for the Screener's perusal during a downgrade. All user inputs and system outputs are moved to GINFO by WCB, but GINFO is written to the GINFO file only when a downgrade request occurs, thus ensuring that the most current information is in the file. WCB maintains GINFO in a wrap around fashion, moving data to GINFO until the end of the buffer is

reached, and then overwriting old data at the beginning of the buffer. When data is written to the GINFO file from GINFO, the start of transference in GINFO is that point at which data was last moved into GINFO.

#### SOM

The System Output Match function (SOM) searches the current context for a match to the current system output. System output contexts may be fixed or variable depending on whether or not the system output that matches them is to be sent to the Screener. System output contexts may also be repeating or non-repeating, which provides an indication of expected system output size. Repeating contexts are used when larger system output is expected (e.g. TLCF messages or TCON skeletons). When a system output is matched to a subcontext, the output is written to GDOWN, and SOM returns a match found condition. If no match is made, SOM returns that fact to FSMGP.

Repeating contexts will contain only one subcontext. If the current context is variable repeating SOM looks for the terminating sequence (from the place in the GHDEL file pointed to by the context) in the system output returned by the Read System Output function (RSO). If the terminating sequence is not found, the searched data is written to GDOWN and another call to RSO is made. This sequence of events is continued until a match is made or a read error occurs, either of which is returned to FSMGP.

If the current context is fixed repeating RSO is called and the system output is matched character-for-character to the subcontext until its end or until the system output is exhausted. In the latter case the system output is written to GDOWN and more system output is read and the process continues. In the former SOM writes to GDOWN and returns that a match has been found.

If the current context is non-repeating there may be several subcontexts to compare against the current system output, and SOM loops for the number of subcontexts. Also, one of the subcontexts could be null, which indicates that any system output is to be accepted. If a fixed subcontext is encountered in the loop SOM attempts an exact match. If a match is found, it is written to GDOWN and SOM exits. If not, the search continues. If a subcontext is encountered that is null, it is saved as a default match. If a subcontext is encountered that is variable, a substring match is attempted on the system output. If a match is found, GDOWN is written and SOM returns to FSMGP. Otherwise the search continues.

At the end of the loop the presence of a null subcontext is determined. If so, a match to the system output is assumed and it is written to GDOWN and SOM returns. If no null subcontext was flagged SOM returns a no match condition.

#### RSO

The Read System Output (RSO) function is responsible for reading WWMCCS system outputs through the Remote Network Processor Emulator (RNPE).



This function must initiate reads from WWMCCS and buffer them so that SOM can attempt a subcontext match. RSO must therefore stop reading when one of the delimiter strings is encountered or a time-out has occurred. The RSO function maintains a local buffer, the input buffer (IB), in which it puts data from RNPE. RSO is passed the delimiter set by SOM (i.e. a pointer to it) so that it may search for these characters in IB. RSO begins searching in IB where it last left off from a previous invocation. RSO searches the buffer until a match is found to one of the delimiter strings or until the end of the system output in IB is encountered. If this is the first invocation of RSO the end of the system output will be found at the outset so that an initial read will occur. The read may result in normal completion with the number of characters read returned, an I/O error from RNPE or a time-out error. RSO returns to SOM when either error condition results. After a read, RSO's buffer pointers are updated and an end of read marker is placed in the buffer indicating the end of the system output, and the search is continued. If a match is found to one of the delimiter strings, pointers to the beginning and end of the system output are returned to SOM.

#### CES

The Check Environment Set function (CES) determines current values for the Screener identification, bandwidth rate and screen mode. Screener identification is used in communicating with SCTP when a downgrade request is made by FSMGP. Since a Screener is not permanently assigned to any given user (the Screener may log off or be reassigned by an EM), FSMGP must know the assigned Screener before a request is made. Bandwidth

rate determines the rate at which messages are allowed to be automatically screened. If this rate is exceeded, FSMGP must change its user's screen mode to 'all', which requires all messages to be screened by the Screener. Screen mode is checked by FSMGP to help it decide whether or not a particular system output is to go to the Screener.

CES must first examine the port fsagn (uniquely identified by the user's identification appended to the port name) to see if the Screener assignment, the bandwidth rate, or the screen mode has been changed. If there is data on the port, the appropriate local FSMGP variables are updated. In addition, if one of the changes involves an environmental value (screen mode or bandwidth), the user's GENV is opened, updated to reflect the change(s), and closed. If there is no data on the port, CES must read the current environment values from GENV and set the appropriate local variables. In either case, CES returns to FSMGP.

#### STS

The Send to Screener function sends a downgrade request to the Screener and takes appropriate action depending on the Screener's decision. The function actually encompasses several logical tasks, but has common entry from several places in FSMGP. If the screen mode mandates all system output must be screened, or if the screen mode is normal and a particular system output is tagged as variable by its matching subcontext, or if the screen mode is normal and the system output is fixed but the bandwidth has been exceeded, the STS function is executed.

When a downgrade is to go to the Screener, STS writes the information in the local buffer GINFO to the GINFO file. GINFO is the downgrade context buffer and is updated with each user input and system output. The GINFO file allows the Screener to view the downgrade material in context to the current user/system interactions. STS then sends a request message to SCTP via the port screq. This port is uniquely identified to SCTP (so it knows where to send the reply) by STS appending the Screener's identifier to the port name. This message contains the name of the user's GDOWN, which is a text file of the information to be downgraded, and the name of the user's GINFO. STS then waits for a reply to the downgrade request. This reply is delivered on the port fsrly, uniquely identified by appending the user's identifier to the port name. If the Screener accepted the material, STS calls the global function GFAUDIT to record that fact on the audit file GFAUDIT. STS then writes the information in GDOWN to the user and returns to FSMGP.

If the Screener rejected the downgrade material, STS audits that fact and sends a message to the user stating the decision. STS must then establish a new context to proceed. This is accomplished by calling a special function that is pointed to by the current context. Since the particular point in the current subsystem is implicit in the current context, the function that it points to will know the appropriate action to take to "pop-up" FSMGP to a level from which to proceed safely. The current context also contains a pointer to another context appropriate to the special function. Therefore STS sets a new context and calls the special function. This function must send an appropriate simulated user

input to the WWMCCS system to resynchronize FSMGP. This is accomplished by sending a canned message to the current WWMCCS subsystem via RNPE. The special function then records the fact that simulated user input has been sent with a call to GFAUDIT.

#### CBR

The Check Bandwidth Rate function determines a dynamic rate of messages automatically screened by FSMGP and compares it to the current setting of the FSM-wide rate. If the dynamic rate is greater than or equal to the set rate, FSMGP returns this fact to FSMGP which must then change the screen mode of its user to "all". If the dynamic rate is less than the set rate, CBR returns this fact to FSMGP and no special action need be taken.

CBR calculates the dynamic rate by keeping track of the time the last system output was sent to the user (the last invocation of CBR), and the number of system outputs automatically bypassed to date. If the time the last system output was sent to the user was sufficiently long ago (defined by some default time set at FSM initiation), a rate is calculated by dividing the number of system outputs that have been automatically screened by the amount of time. If this rate is greater than the system-wide set rate, CBR returns that value, otherwise CBR returns nothing. In either case CBR must reinitialize its counters before returning. If the time was not sufficiently long ago to calculate a rate, no counters are changed and CBR returns nothing. On its initial invocation, CBR establishes its counters and returns nothing.

## LOGOFF

The LOGOFF function performs clean-up activities when the user logs off the WWMCCS system. It must make sure that the assignment daemon knows about the logoff. LOGOFF is invoked when the user indicates he is logging off, for example entering 'bye'. When this occurs, a pointer in the context with the user's input will direct FSMGP to execute LOGOFF. LOGOFF communicates with FASSD over its port FAPORT, informing it that the user has logged off.

### 3.2.2 Screener Trusted Program (SCTP)

The Screener Trusted Program (SCTP) interfaces directly with the FSM Screener when the Screener logs onto FSM. SCTP is responsible for displaying downgrade information to the Screener in order for the Screener to determine if the information to be written down to the low security domain of FSM contains sensitive high security level data. This information is presented to the Screener one screen at a time. Any time during this data review sequence the Screener can reject the data for downgrading. However, the entire downgrade file must be displayed to the Screener before he can accept it for downgrade. An additional downgrade context information file is also available to the Screener in order to assist in determining if the downgrade information contains sensitive data. This downgrade context file contains two screen fulls of information reflecting the user/system dialogue just prior to the downgrade data. This data can be perused by the Screener following the inspection of the downgrade information. During the perusal of the downgrade

context file, the Screener can accept/reject the downgrade request.

3.2.2.1 Global Data Usage. SCTP utilizes the following FSM global data structures in performing its function:

GLF SCTP updates the logon file (GLF) when a Screener logs onto FSM.

GDOWN SCTP reads in the downgrade file (GDOWN) and displays its contents to the Screener for review.

GINFO SCTP reads in the downgrade context information file (GINFO) and displays its contents to the Screener for review.

Audit Requests SCTP builds audit trail requests whenever the Screener logs onto FSM and when the Screener begins reviewing a downgrade request.

3.2.2.2 Local Data Usage. SCTP uses no pertinent local data structures.

3.2.2.3 General Processing Flow. The Screener Trusted Program (SCTP) is invoked when a FSM Screener logs onto FSM. There is a separate copy of SCTP for each Screener logged onto FSM. As soon as SCTP is initiated it locks, opens and reads in the Global Logon File (GLF), updates its contents reflecting the Screener logon, writes GLF out, then closes and unlocks the file. SCTP audits the Screener logon event via the global function GFAUDIT.

Following Screener registration into GLF, SCTP creates its screening request port based on the Screener's user ID. This port will be utilized in receiving screening requests from the zero or more FSM Guardian Processes assigned to this Screener.

Next Sctp checks the screening request port for possible screen requests. If requests are not present then Sctp outputs to the Screener a set of prompts stating that there are no pending screening requests and he may log out if he wishes. Then Sctp checks for both Screener logout requests or potential screening requests. If neither event occurs for some predetermined time Sctp again outputs no screening request/logout prompts to the Screener. This sequence continues indefinitely until either the Screener logs out or a screening request appears.

If the Screener logs out, Sctp deregisters the Screener from GLF by writing a message to FASSD via its port (faport).

If a screening request is present then Sctp notifies the Screener with the prompt options to continue with the screen request or logout. Assuming the Screener decides to continue with the screen request, Sctp then reads in the downgrade file (GDOWN) indicated in the request port message and displays the first page to the Screener. Following this page of data are one of two possible prompts: If this is not the only page of data then Sctp prompts the Screener with a reject/continue option; if this is the first and last page of downgrade information, Sctp prompts the Screener with accept/reject/continue options. For the first situation, the Screener continues screening the downgrade information by continuing for each subsequent page. Following complete display of the downgrade file, the Screener has the option of viewing a downgrade context information file (GINFO). This is accomplished by continuing past the downgrade file. For each page of context data, Sctp prompts the

Screener with an accept/reject/continue option. If the Screener continues through and past the context information file, Sctp will cycle back to the downgrade file, ad infinitum.

During the above sequence the Screener will eventually either accept the downgrade or reject it. If the Screener accepts the downgrade request Sctp requests that he confirm his action. If the Screener does not confirm the accept, Sctp continues displaying information at the point where it left off. If the Screener does confirm the accept action, Sctp notifies the appropriate FSMGP through the screening reply port (fsrly-i). Then Sctp notifies the Screener that the confirmation has been acted upon and continues at the beginning of its cycle looking for another screening request.

If the Screener decides to reject the downgrade, Sctp requests that he confirm the rejection. If the Screener does not confirm the rejection, Sctp continues displaying data. If the Screener confirms the rejection, Sctp notifies the appropriate FSMGP of the rejection through the screening reply port (fsrly-i). Then Sctp notifies the Screener that the rejection has been acted upon and then proceeds to obtain another screening request.

**3.2.2.4 Major Functions.** Following are a set of salient functions important to Sctp operation.

SCUGLF

The Update GLF function is responsible for processing the Global Logon



File (GLF). This function operates on an input parameter specifying the Screener's logon name. It locks GLF (using global function GFLOCK). The Screener's GLF record is sought and, if found, is checked to see if this Screener is currently logged on. If so, the Screener is informed and his SCTP terminates. If the found record indicates an inactive record (not logged-on), SCTP marks it active and updates GLF with it (the file is still write-locked). If no GLF record is found for this Screener, he is informed and his SCTP terminates. The GLF file is unlocked following this sequence, regardless of the outcome of the logon.

#### SCPROMPT

The Screener Prompt Function is responsible for determining the appropriate user prompt based on the current state of SCTP. Prior to outputting the next appropriate prompt, SCPROMPT flushes the Screener's terminal input in order to remove superfluous characters that the Screener may have entered. Then SCPROMPT determines the prompt on the following SCTP states. If the current SCTP state is proceed, then valid prompts are "logout (l)" or "continue (<CR>); if the current state is continue, then valid prompts are "continue (<CR>)" and "accept (a)", and "reject (r)". If the Screener has not seen all downgrade information and the state is continue, then "continue (<CR>)" and "reject (r)" are valid prompts. If the state is accept, then "confirm-yes" and "confirm-no" are valid prompts. If the state is reject, then "confirm-yes" and "confirm-no" are valid prompts.

### 3.2.3 FSM Assignment Daemon (FASSD)

The FSM Assignment Daemon (FASSD) is responsible for automatic WES user/Screeners assignments within the FSM. The Environment Manager (EM), via the EMP, can manually reassign users and Screeners as desired. However, since the EMP is not always expected to be running, FASSD makes all initial assignments as WES users and Screeners log on and off.

3.2.3.1 Global Data Usage. FASSD uses the following global data structures:

GLF        FSM global logon file.  
GAEV       Current audit event files.  
FSAGN-i    FSMGP assignment reply port.  
FAPORT     FASSD assignment request port.

3.2.3.2 Local Data Usage. FASSD uses no pertinent local data structures.

3.2.3.3 General Process Flow. FASSD is a daemon which is initiated at FSM system startup. FASSD reads its input port, faport, for an IPC message and blocks on it if empty. If there is at least one port message in faport, FASSD reads the message and checks the message type.

IPC messages from EMP can only be WES user/Screeners reassignments (SCCNG) followed by the WES user ID and Screener ID to reassign. FASSD reads the Global Logon File (GLF) searching for the WES user ID. When the user's record is found, the Screener ID for this WES user is changed to reflect

the new Screener that the EM reassigned. The record is updated in GLF (after GLF is locked) and FASSD generates an audit event indicating a WES user/Screener assignment has taken place. FASSD checks its port for more messages and if none, blocks.

IPC messages originating from SCTP can either be Screener Logon (DRRDY) or Screener Logoff (DRLGO), followed by the Screener ID and (for DRRDY messages only) the Screener's UNIX terminal ID. If the port message is DRRDY, FASSD may make WES user/Screener assignments if this is the only Screener currently logged onto FSM. If the port message is DRLGO, FASSD searches the GLF for a record with the FSM ID the same as the passed Screener ID and marks the record inactive. Then, FASSD searches GLF looking for active WES users which are assigned to this Screener. If there are any, FASSD uses its assignment algorithm to reassign WES users to other Screeners. If there are no other Screeners available, FASSD sends a port message to the FSMGP of each WES user denoting that no Screeners are available (SCNAV). FASSD generates an audit event when the Screener logs off and whenever a WES user/Screener assignment occurs.

IPC messages from FSMGP can be either User Logon (UAREQ) or User Logoff (UALGO) followed by the user ID and terminal ID. Similar to SCTP message types, FASSD searches GLF for a user ID that matches the passed user ID. In the UAREQ case, FASSD assigns the WES user to a logged-on Screener, if any. FASSD notifies FSMGP of the assignment by sending a Screener Assigned (SCASS) message and the assigned Screener's ID to the FSMGP's port, fsagn-i. If no Screener is available, FASSD sends the SCNAV message to

this port. An audit event is generated if an assignment occurs. In the UALGO case, FASSD marks the WES user's GLF record inactive. FASSD generates an audit event indicating that the WES user has logged off of FSM.

3.2.3.3.1 Specification of Assignment Algorithm. Four cases need to be considered with the assignment algorithm:

1. Screener Logs On
2. Screener Logs Off
3. User Logs On
4. User Logs Off

3.2.3.3.1.1 Screener Logs On. FASSD makes no assignment changes in this case. Even though there may be a heavy load on another Screener, no users are automatically assigned to the new Screener, in the interest of user/Screener stability. The new Screener will be the prime candidate for future assignments however.

3.2.3.3.1.2 Screener Logs Off. This is the most serious case since it is likely that at least one WES user, and possibly more, will need reassignment. If no other Screeners are available, the port message No Screeners Available (SCNAV) is sent to each FSMGP's fsagn-i port. FSMGP allows the user to proceed until a Screener is needed, at which time the

user has the option of logging off or waiting for either a Screener assignment or screen mode change (via the EMP) to occur.

If other Screeners are available, FASSD chooses the Screener with the smallest current load (fewest users assigned) to reassign the user to. This step may be repeated multiple times in order to reassign all WES users who were assigned to the logged-off Screener. Each time, the Screener with the smallest current load is chosen. In the case of ties (multiple Screeners with the same load), FASSD chooses the Screener who most recently logged on. This is done because the most recently logged-on Screener will probably remain logged-on longer than the other Screener(s). This scheme requires that FASSD record the Screener logon DTG in a local data structure (following a DRRDY port message from a SCTP).

3.2.3.3.1.3 User Logs On. Similar to the previous case, FASSD chooses the Screener with the smallest current load to assign the WES user to. Ties are resolved as stated above. This case will require at most one assignment. As before, there may be no Screeners available.

3.2.3.3.1.4 User Logs Off. FASSD makes no assignment changes when a WES user logs off. The reasoning follows that discussed in "Screener Logs On". The EM may manually reassign users and Screeners; however, in the light of user/Screener stability, FASSD makes no assignments.

### 3.2.4 FSM Audit Daemon (FAD)

FAD logs audit events in the current audit files based on requests in the audit queue. Audit requests are processed on a strict FIFO scheme. Audit request files are opened, processed and unlinked as the audit event specified is actually logged in the audit files.

#### 3.2.4.1 Global Data Used. The following global data is used by FAD:

Audit Request Files Contains audit events to log.

AUDITQ Directory contains all audit request files.

Current Audit File-Pair Contains logged audit events.

GAUD Audit description file.

GAUD-LOCK Write-lock mechanism for audit files.

- 3.2.4.2 Local Data Used. FAD uses the UNIX file status buffer returned by the "stat" call.

3.2.4.3 General Process Flow. Due to the large volume of audit events, FAD is implemented as a daemon, rather than a callable process or function. As such, it has no input or output parameters. All data that FAD uses in logging an audit event is either contained in the audit request file(s) or is obtained by FAD. FAD will sleep for 30 seconds, wake up and examine its queue, "/fsm/fsat/auditq", for pending requests. If there are no file entries in the Audit Queue directory, FAD returns to the sleep state and repeats this process again upon waking. If there are pending audit requests, FAD must find their filenames and Date-Time Group (DTG) of last modification. The filenames and DTGs of last modification

are entered in a list and are sorted on the DTG using the UNIX qsort function. The filename with the least recent last modification (the oldest file) appears first in the list, followed by the second oldest file and so on.

FAD locks the audit files (GAUD) for writing by calling GFLOCK. Then, it opens the current audit file-pair and the next oldest audit request file. FAD increments the pointer into its sorted filename list, so on the next pass it will pick up the next oldest file. After reading the event record and saving it, FAD reads the text record(s), if any, and copies them byte-by-byte to the current Text file; remembering the beginning location of the text (OFFSET) and keeping a count of the number of bytes transferred. FAD sets up the event record, updates it (with OFFSET and LENGTH) if necessary, and writes it to the current Event file. The event has now been logged, so FAD closes the current audit file-pair. FAD signifies successful event logging by closing and unlinking the audit request file just processed from the audit queue.

FAD must update the Audit Description (GAUD) file, as well, with the new "last entry" DTG for the current audit file-pair. FAD opens GAUD (the audit files are still write-locked), locates the last record in the file and reads it into a small buffer. The DTG of last entry is updated and the record is written back to the GAUD file. FAD closes and unlocks the audit files for writing by calling GFUNLOCK.

If there are any files left in the sorted list, FAD repeats the sequence described above until there are no files left in the list. Then, FAD

re-examines the audit queue and if there are any files, continues as described above. Otherwise, FAD will sleep before repeating this entire process.

### 3.2.5 Environment Manager Program (EMP)

The Environment Manager Program (EMP) permits monitoring and alteration of the FSM operating environment. EMP performs manual WES user/Screeners assignments, profiles WES user's status and modifies FSM screen modes and bandwidth thresholds. The EMP is run by the Environment Manager (EM).

#### 3.2.5.1 Global Data Usage. EMP uses the following global data:

GENV System (default) and user's environment files.

GLF FSM global logon file.

GPROF User profile file.

#### 3.2.5.2 Local Data Usage. The following local data is used by EMP:

1. Screener list - a variable-length list whose entries each contain a one-byte Screener ID and an eight-byte Screener logon name.
2. User list - a variable-length list whose entries each contain a one-byte user ID, a one-byte Screener ID (who the user is currently assigned to) and a twelve-byte WWMCCS user ID.



3.2.5.3 General Process Flow. When invoked by the EM, EMP checks the Global Logon File (GLF) to verify the logon name that was entered and to ensure that the EM is not currently logged on. Only one EMP can exist at any given time, and therefore only one EM record can be defined in GLF. If the EM name is valid and not logged on, EMP marks the record as active and updates GLF (after locking it for writing). Then, the EMP help file is displayed which briefly describes available EMP commands. Following this display, the EM is placed at the EMP's command level, signified by an asterisk on the EM's terminal. EMP commands are valid at this point. See Appendix B for detailed command information.

If the assignments command is entered, the EMP prints the current list of user/Screener assignments. This is done by reading each Global Logon File (GLF) record and checking the Logged Bit Flag (DLOGDON) for logged-on users. Records of users that are not logged onto FSM are not considered. For each active (logged-on) record, EMP also checks whether this user is a WES user or a Screener. If the active record indicates a WES user, the FSM user ID, user's WWMCCS user ID and the Screener ID are stored in the user list. If the active record indicates a Screener, the Screener ID and the Screener's logon name are stored in the Screener list. After all GLF records have been read, EMP checks each record in the Screener list. The Screener ID of each record in the Screener list is compared with each Screener ID in the user list. When a match is found (there may be more than one), the Screener user logon name from the Screener list is printed on the EM's terminal, followed by the user's logon name from the user list. Each record in the user list is compared

with records in the Screener list. For each Screener List record, if a match occurs a flag is set. After all user list records have been compared with a Screener list record, the flag is checked. If it was never set, the Screener associated with this Screener list record is not assigned to any users. This information is important for future EM-directed WES user/Screener reassignments, hence the Screener's user logon name from the Screener list is printed on the terminal followed by the string "- none assigned -". When the Screener list is exhausted, EMP returns to command level.

If the EM enters the display command at EMP command level, EMP prints out current system/user environment values. Display entered without arguments causes the current system environment values to be listed, followed by the current environment values of every logged-on WES user. If optional names follow the display command, they must be names of logged-on WES users or the keyword "SYSTEM". This causes EMP to only display those user's status. If no arguments followed the command name, EMP reads the system Global Environment (GENV) file and displays the bandwidth threshold and the WWMCCS subsystem screen modes. Additionally, EMP reads each GLF record and keeps counts of the number of logged-on WES users and Screeners. This information is printed out on the terminal. Following the system display, each GLF record is read. For each logged-on WES user, the user ID is used to form a filename of the form "/fsm/usr/user-i/genv". This file contains the user's current environment values. EMP reads it and displays the information on the EM's terminal. When all GLF records have been read, EMP returns to the command

level. If optional names follow the display command, each name is compared with the string "SYSTEM". If a match occurs, the procedure described above for displaying system environment values is performed. If the name does not match "SYSTEM", then each logged-on WES GLF record is checked with the user name. If there is no match for a name, EMP warns the EM and requests a replacement user name. If there is none, EMP continues with the next name following the display command, otherwise, it repeats the process for the replacement user name. Each name is processed identically in sequential fashion. The WES user's GENV file is read and the contents formatted for display. When there are no more names remaining, EMP returns to the command level. The display command has a limit of five user names and if this number is exceeded, EMP prints a message and ignores the remaining names.

If the history command is entered at the command level, EMP fork/executes the Audit Trail Display Program (ATDP). This command enables the EM to view previous audit events. The audit event information may aid the EM in making a FSM environment change. When the EM terminates ATDP (by using the quit command), EMP returns to the command level. See Appendix D for detailed ATDP command formats.

If the modify command is entered at the command level, EMP parses and saves any optional arguments. The modify command has both long and short command forms and depending upon which is used, various displays and prompts appear. In the case of no command arguments, EMP prints a menu of valid modification actions and then prompts the EM for the action

desired. The selected action then causes a new menu to be displayed and the EM is prompted to enter the new environment value(s). When finished with this modification action, the EM is returned to the initial menu which details modification actions. This was done so that the EM can modify multiple values with one modify command. If command arguments are present, they are parsed syntactically and errors cause EMP to re-prompt for the bad value(s). When the modification has been made, in this case, the EM is returned to the command level. This is an important distinction between the long and short command forms. The short form enables multiple changes to be made with a single modify command, with the EM being prompted from menus. The long form validates the command arguments, modifies the desired value(s) and returns to the command level. Except for this external distinction, the two command forms operate identically. If the system-wide environment values are modified, EMP writes the new values in the system Global Environment (GENV) file. The values in this file are given to each WES user at logon. To alter currently logged-on user's values, a port message (BRCNG or SMCNG) is sent to each FSMGP noting that an environment value has changed and indicating the new value(s). System-wide values that can change include bandwidth threshold and subsystem screening status. Notice, EMP does not update each WES user's GENV file. Following the system-wide change, EMP generates an audit event which details the change and new values. If a specific user's environment values are modified, EMP writes a port message to the user's FSMGP, via the fsagn-i port, noting the change and new value(s). User environment values, like system-wide values, which can be

altered are bandwidth threshold and subsystem screening modes. Again, the EMP does not alter the WES user's GENV file. EMP generates an audit event which details the change and new values. The final type of modification that can occur is WES user/Screeners reassignment. EMP verifies that both the WES user and Screener specified by the EM are logged onto FSM. If they are not, EMP messages and re-prompts for user names. If both names are valid and logged-on, EMP sends a port message (SCCNG), via the faport, to the FSM Assignment Daemon (FASDD) indicating the user ID and Screener ID to reassign. There is no communication between the user's FSMGP and EMP in this case. EMP returns to command level after the port message has been sent. Notice, EMP does not directly reassign WES users and Screeners and it does not generate an audit event.

If the profile command is entered at the command level, EMP displays the WES user's profile. Profile followed by no optional command arguments causes EMP to read the GLF and for each logged-on WES user, to form the filename "/fsm/usr/user-i/gprof". This file contains the last wait condition for this user. Additionally, the file contains the WES user's terminal ID, WWMCCS subsystem and DTG of the wait. This information is displayed, one line per user, on the EM's terminal. When the GLF file has been exhausted, the EMP returns to command level. If optional command arguments follow the command, EMP checks the GLF for a match between each argument and GLF WES user names (if the argument is longer than one byte) or GLF terminal IDs (if the argument is a single byte). If no match is found, the EMP prints a warning message and the EM is prompted for a replacement name. If a match is found, the user ID of the GLF

record is used to form the filename shown above and the file contents are formatted and displayed on the EM's terminal. As with the display command, there is a limit of five names following the command and remaining names are ignored with a message.

The final EMP command is one to terminate the program. Since EMP has no input ports and automatic WES user/Screeners assignments are performed by the FSM Assignment Daemon (FASD), the EMP does not always need to be running during FSM operation. If the EM enters the quit command, EMP exits. The EMP can be run at any future time the EM desires.

#### 3.2.6 Audit Trail Display Program (ATDP)

ATDP performs all formatting and display of generated audit events. It also is used to prevent the current audit file-pair from growing to an unmanageable size. ATDP is invoked by using the HISTORY command in the Environment Manager Program (EMP).

##### 3.2.6.1 Global Data Usage. The following global data is used by ATDP:

Current Audit File-Pair Contains logged audit events.

GAUD-LOCK Audit files lock mechanism.

Saved Audit File-Pair(s) Contains previously logged audit events.

GAUD Audit description file.

3.2.6.2 Local Data Usage. ATDP does not use any pertinent local data.

3.2.6.3 General Process Flow. After ATDP has prompted the EM for command input (signified by ATDP printing a ">" on the EM's terminal) and has read it, argument parsing begins. This is a major phase of ATDP since at least one command has several optional variables, with many variables also taking several optional arguments. The only restrictions on command input are argument ordering. See Appendix D for detailed command syntax and descriptions. The ATDP command name is always the first argument. In the LIST command, the flag variable is always second (if present), followed by the event-name (if present) and the time-range (if present). Several non-fatal errors may be detected during this parsing phase and if an unrecognizable argument is decoded, ATDP prompts for the value again after printing a warning message detailing the problem.

If the command to ATDP is END, ATDP write-locks the audit files by calling GFLOCK. This is necessary since ATDP renames the current audit file-pair and updates GAUD. ATDP renames the current audit file-pair from "current" to the DTG of the END operation. For example, ENDing the current audit file-pair on April 2 at 12:05:30 causes a rename from "current.e" to "0402120530.e" for the Event file and a rename from "current.t" to "0402120530.t" for the Text file to take place. ATDP then updates GAUD (audit files are still write-locked) to change the filename of the old current file to the new DTG and to create a new current record with the filename of current and the DTG of file creation. Then, ATDP unlocks the audit files by calling GFUNLOCK.

If the command is QUIT, ATDP exits back to the Environment Manager Process from which it was fork/executed.

If the command is LIST, ATDP checks the second argument for a flag value. If it is a flag argument, it must be preceded by a "-". The flag indicates method and destination of the output. Methods include paging and no paging. Destinations include terminal and printer. If the second argument is preceded by a "-" and the remaining character is not a valid flag argument, a warning message is printed and the flag is ignored.

ATDP checks the next argument, if any, for an event name keyword. If a valid event keyword is found, and there is no valid event type following it, ATDP prompts the user for a list of event types. Otherwise, ATDP decodes the event type and sets the appropriate internal flags. Event types include FSM user names, WWMCCS subsystem names and event codes. This argument serves to limit ATDP output to a selected subset.

ATDP checks the next argument(s), if any, and expects to find valid time ranges or time keywords. A time range is a numeric string indicating month, day, hour and minute or a time keyword. If the remaining argument(s) are not valid time ranges, a warning message is printed. ATDP also reprompts the user for the correct time range. Time ranges, like event types, serve to limit the output to a selected subset of events.

Once all arguments for LIST have been parsed, validated and internal program flags set, ATDP reads the GAUD file from the beginning, searching



for the first audit file-pair whose date of last modification is greater than the LIST start-time. By default, the start-time is midnight on the current day. Once a file has been found, ATDP opens it for read-only. Each event is read from the Event file and its FEVTIM (DTG of the event) is compared with the start- and end-times to list. If the record is within the specified time range, ATDP checks the event name condition(s), if any. By default, all event types are chosen. If the event record satisfies the event name condition(s), it is printed out. The event's text, if any, is read and written to the output destination in a byte-by-byte fashion. When the current event is done, ATDP reads the next Event file record. At end of file, ATDP gets the next audit file-pair in the GAUD file. ATDP stops looking for next event records (files) when the DTG of the event is greater than the end-time.

#### SECTION 4

#### QUALITY ASSURANCE

This section is not applicable to this document.

**SECTION 5**

**PREPARATION FOR DELIVERY**

**This section is not applicable to this document.**

## SECTION 6

### NOTES

This section is not applicable to this document.

## APPENDIX A

### FSM CONTEXT TABLES

#### A.1 OVERVIEW

The low-level context tables which are used by the FORSCOM Security Monitor (FSM) are produced from high-level files created with the UNIX text editor, ed. A C program named Conbuild exists which makes the transformation from the human-oriented, high-level files to the machine-oriented, low-level files.

Using Conbuild offers many advantages in context table construction. All labels used are symbolic and thus do not require alteration if the file is changed. Non-printing ASCII characters can be easily entered and visually checked. In addition to generating correct low-level tables, Conbuild checks and verifies high-level values. Errors detected in the high-level tables can be quickly pinpointed and corrected.

Conbuild accepts three high-level files and produces three low-level files: context, halting delimiter and text. The formats of the high-level files are detailed below. The halting delimiter and text files make use of special symbols to visualize non-printing or hard to see characters. Special symbols are character strings preceeded and followed

by a vertical bar ("|"). Conbuild translates the special symbols into the appropriate character sequence. Special symbols include:

	one blank
bell	terminal bell
cr	carriage return
ctl-A	control-A
lf	line feed
nl	new line (carriage return, line feed)
	null byte

## A.2 CONTEXT

The high-level context file follows this format:

Cs	Cp	Dh	bf	ss	[pf]	[!comment]
Ts	Cn	bf	gs	gl	[sf]	[!comment]
			.			
			.			
Ts	Cn	bf	gs	gl	[sf]	[!comment]

The first line of every context is the context header. The fields are:

1. Cs -- The context label

This symbolic label serves to identify this context. It can be used by other contexts (as a pop-up reference, following an abnormal match) and subcontexts (as a go-to context, following a match).

2. Cp -- Pop-up context

If, during normal context processing, an input is received from the user or system which cannot be matched in any of the

subcontexts, a "pop-up" is effected. This label serves to symbolically identify the context to process next, in the event of a pop-up.

3. Dh -- Halting delimiter set

The halting delimiter set is identified by this symbolic label. It is an offset into the halting delimiter file where the desired delimiter set can be found.

4. bf -- Context header bit flags

The header bit flags are positional and represent the following:

i. Source of next input

This bit flag can take the value "U" (user input), "S" (system output) or "B" (both). This flag identifies the location of the next input to process. In the degenerate case of both, polling is done and the first input received is processed.

ii. Length of next input

This bit flag can take the value "R" (repeating) or "N" (non-repeating). In the case of multi-line system outputs, such as bare TCON skeletons,

multiple reads will need to be performed to obtain the entire output. This occurs because the read function stops reading when a halting delimiter is encountered. One logical system output may contain many halting delimiters.

iii. Matching of next input

This bit flag can take the value "F" (fixed) or "V" (variable). The flag identifies the type of matching to be performed, during repeating reads only. This flag does not supercede the subcontext fixed/variable bit flag.

iv. Subsystem entry to audit

This bit flag can take the value "A" (audit) or "N" (no audit). The flag, if "A", causes an audit event to be generated that records the fact that a new WWMCCS subsystem was entered.

5. ss -- Subsystem name

This field contains the name of the WWMCCS subsystem which this context item pertains to. The field is used for audit purposes when the audit bit flag is set.



6. pf -- Pop-up function name

This optional field, if present, contains the name of a C function to execute which will simulate user input, so as to effect a pop-up to the pop-up context.

7. !comment

An exclamation mark followed by anything can appear at the end of any context header line. It is ignored by all programs. All comments are optional.

Each line of input following the context header is a subcontext item that is part of the context item. The fields are:

1. Ts -- Text string to match on

This symbolic label identifies the text string to use when matching input. It is an offset in the text file of a null-terminated string.

2. Cn -- Next context if match made

If this subcontext's text matched the input, use this symbolic label as the next context to process. It is an offset in the context file of a context item.

### 3. bf -- Subcontext bit flags

The subcontext bit flags are positional and represent the following:

#### i. Type of matching to perform

This bit flag can take the value "F" (fixed) or "V" (variable). The value determines the type of matching to perform on the input received. A fixed match compares the input against the text string byte-by-byte. A variable match compares the text string against the input, looking for a match anywhere.

#### ii. All or partial match

This bit flag can take the value "S" (substring) or "N" (no-substring). Used in combination with the previous bit flag, it denotes whether the entire input is to be matched against, or only part of it. Fixed no-substring requires that the input and the text string be of equivalent sizes, order and each byte matches (i.e. an exact match). Fixed substring requires that a small part of the text string match the input exactly. Following a match, the text string is used and not the input that was received. Variable no-

substring involves no matching and is otherwise termed a "null subcontext". This can be thought of as a "wildcard" and will match any input. Variable substring is similar to fixed substring, with the exception that the input is used, rather than the text string.

iii. Special function necessary

This bit flag can take the value "F" (function required) or "N" (no function required). This flag denotes whether the optional field "sf" will be present or not.

iv. Validity of input

This bit flag can take the value "V" (valid input) or "I" (invalid input). Following a match between the received input and the text string, this bit is checked and if not set, the input is not allowed to proceed due to its invalidity. This occurs in subsystems which have commands that FSMGP does not allow to be used. For example, the TCON OLD command is a valid WWMCCS input, but is not part of the subset of WWMCCS operations that FSM allows.

v. Source of input

This bit flag can take the value "U" (user input) or "S" (system output). In the event that the header bit flag was "Both", this bit flag identifies whether this subcontext pertains to the received input or not.

vi. Disposition of input following match

This bit flag can take the value "E" (end of current input) or "B" (beginning of next input). Its use is limited to variable substring matches and serves to identify whether the substring matched is part of the current input, to be downgraded, or is the beginning of the succeeding input. The flag will normally be "E", signifying that the substring matched is to be downgraded with the rest of the input received.

4. gs -- Start of fixed substring match

This field pertains only to fixed substring matches and in that event, gives an offset into the text string to use in matching. The length of the substring, since it is not null-terminated, is given by the next field, "gl".

5. gl -- Length of fixed substring match

This field pertains only to fixed substring matches and in that event, gives a length from the start of the substring to use in matching. In the event that the substring defined by these two fields is matched, the input received is discarded and the entire text string is used as the input.

6. sf -- Special function name

This optional field, if present, contains the name of a C function which can be to perform special actions in regard to the input received, following a successful match. Initially, this field only pertained to user input and was limited to validating complex user inputs so as to reduce the possible number of error messages received. Since that time, the concept has been generalized to include system output as well. The special function can implement any special actions that are not normally performed during the FSMGP's processing loop.

7. !comment

An exclamation mark followed by anything which appears at the end of a subcontext input line is regarded as a comment and is ignored by all programs. It is optional.

Context items, which include the single context header followed by zero or more subcontext items, are delimited from each other by a blank line.

### A.3 HALTING DELIMITER

The halting delimiter file follows this format:

```
Ds      d1^d2^ ... di^^  
        or  
Ds      d1^  
        d2^  
        .  
        .  
        .  
        di^^
```

Each halting delimiter set contains one or more halting delimiters. Halting delimiters are delimited from each other by a special character that is called a "pseudo-null". This character will be recognized as a meta-symbol and translated to a null byte everywhere it is encountered. The pseudo-null character is a carret ("^"). This indirection is necessary due to the editor's inability to insert nulls in a file. Each halting delimiter set is doubly null-terminated.

The fields in a halting delimiter set are:

1. Ds -- The delimiter set label

This symbolic label is used in the context file as a reference to a delimiter set.

## 2. dj -- The halting delimiters

Each halting delimiter is null terminated and can contain any meaningful characters that are legitimate halting delimiters. Typically, halting delimiters will be new-lines or carriage returns.

A halting delimiter set can be entered on a single line or on multiple lines, as desired. It is best NOT to split a halting delimiter across new-lines, however, it will be interpreted correctly. In the event multi-line input is used, following the new-line there must appear a tab. This serves to denote that the set is not yet terminated, as well as providing a visual mnemonic to that effect.

Halting delimiter sets are delimited from each other by a blank line.

### A.4 TEXT STRING

The text string file follows a similar format to the halting delimiter file. The format is:

```
Ts      ttttttt ... ttt~  
        or  
Ts      ttt  
        tttt  
        .  
        .  
        .  
        ttt~
```

Each text string contains zero or more characters and is terminated by the "pseudo-null" character, as described above. Each text string requires a unique label and is delimited from other text strings by a blank line.

The fields are:

1. Ts -- The text string label

This symbolic label identifies the text string and is used in the context file. It must be unique.

2. t -- The text string

The text string must be null-terminated and can be entered on a single input line or on multiple lines. If using multi-line input, following a new-line there must be a tab. Similar to the halting delimiter file, this provides an indication that the text string is not terminated.

#### A.5 SOURCE CONTEXT TABLES

Following is a listing of the source context files for the teletypewriter and wang terminals which were used during the PROUD SPIRIT exercise.



### A.5.1 Source Context File (GCONT)

The context file which follows has had its comments truncated to eighty columns in order to fit onto the page. All essential data is otherwise intact.

C1	C999	D1	SRVN	LOGON	outnup ! begin tty/wang logon
T2	C10	VSNVSE	0	0	! "1316400" (line number)
T1	C10	VNNVSE	0	0	! (anything else)
C10	C999	D20	SNVN	LOGON	outnup ! system prompts for subsystem
T10	C20	VSNVSE	0	0	! "PROGRAM NAME --"
C20	C999	D40	UNFN	LOGON	outnup ! user enters subsystem
T20	C30	FNNVUE	0	0	! "tss"
T21	C200	FNNVUE	0	0	! "tlcf"
C30	C999	D30	SNFN	LOGON	outnup ! Begin TSS logon
T30	C40	FNNVSE	0	0	! "TERMINAL "
C40	C999	D1	SNVN	LOGON	outnup ! system identifies terminal
T40	C50	VSVFSE	0	0	termid ! "nn" (terminal id)
C50	C999	D50	SNFN	LOGON	outnup ! system prompts for userid
T50	C60	FNNVSE	0	0	! "USERID\$PASSWORD ... "
T51	C999	FNFVSE	0	0	locked ! "LOCKED" (terminal is locked)
C60	C70	D40	UNVN	LOGON	outnup ! user enters userid
T60	C60	VSNUIE	0	0	! <uid>\$<pw>/scc/scc [invalid]
T61	C90	VSVFUE	0	0	getuid ! <uid>\$<pw>
C70	C999	D40	SRFN	LOGON	outnup ! system re-prompts for userid
T50	C80	FNNVSE	0	0	! "USERID\$PASSWORD ... "
C80	C999	D40	UNVN	LOGON	outnup ! user enters userid (last chan
T60	C80	VSNUIE	0	0	! <uid>\$<pw>/scc/scc [invalid]
T61	C90	VSVFUE	0	0	getuid ! <uid>\$<pw>
C90	C999	D1	SRFN	TSS	outnup ! system prompts for ident
T90	C100	FNNVSE	0	0	! "IDENT?"
C100	C999	D40	UNVN	TSS	outnup ! user enters identifier
T100	C110	VSNVUE	0	0	! <ident>
C110	C999	D110	SNFA	TSS	outnup ! system responds to ident

T110	C120	FNNVSE	0	0	! "CLASSIFICATION OF YOUR OUTPUT?"
T110	C120	VSNVSE	0	0	! "CLASSIFICATION OF YOUR OUTPUT?" +
T111	C120	FNNVSE	0	0	! "classification of your output?"
T111	C120	VSNVSE	0	0	! "classification of your output?" +
T90	C100	FNNVSE	0	0	! "IDENT?" (previous one bad)
C120	C999	D40	UNFN	TSS	outnup ! user enters output class
T121	C130	FNFVUE	0	0	getcls ! uzz
T122	C130	FNFVUE	0	0	getcls ! ufo
T123	C130	FNFVUE	0	0	getcls ! czz
T124	C130	FNFVUE	0	0	getcls ! szz
T125	C130	FNFVUE	0	0	getcls ! swp
C130	C999	D130	SNFN	TSS	outnup ! system responds to output cla
T130	C140	FNNVSE	0	0	! "CLASSIFICATION OF FILES ... "
T131	C140	FNNVSE	0	0	! "classification of files ... "
T132	C110	FNNVSE	0	0	! "ILLEGAL CLASSIFICATION CODE"
T133	C110	FNNVSE	0	0	! "illegal classification code"
C140	C999	D40	UNFN	TSS	outnup ! user enters file class
T121	C150	FNNVUE	0	0	! uzz
T122	C150	FNNVUE	0	0	! ufo
T123	C150	FNNVUE	0	0	! czz
T124	C150	FNNVUE	0	0	! szz
T125	C150	FNNVUE	0	0	! swp
C150	C150	D150	SNVN	TSS	outnup ! system responds to file class
T150	C160	FNNVSE	0	0	! "SYSTEM ?" (only)
T151	C160	FNNVSE	0	0	! "SYSTEM ?" (after pop-up)
T150	C160	VSNVSE	0	0	! "SYSTEM ?" (preceded by message)
T132	C130	FNNVSE	0	0	! "ILLEGAL CLASSIFICATION CODE"
T133	C130	FNNVSE	0	0	! "illegal classification code"
C160	C150	D40	UNVN	TSS	outnup ! user enters TSS subsystem
T160	C1000	FNNVUE	0	0	! acce (ss)
T161	C1000	FNNVUE	0	0	! acces (s)
T162	C1000	FNNVUE	0	0	! access
T170	C7000	FNNVUE	0	0	! list (with no arguments)
T163	C7000	VSFVUE	0	0	vumcl ! list
T164	C6000	FNNVUE	0	0	! sios
T165	C2000	FNNVUE	0	0	! tcon
T166	C9000	FNNVUE	0	0	! wwdm (s)
T167	C9000	FNNVUE	0	0	! wwdms
T168	C990	FNNVUE	0	0	! bye (logoff)
T171	C7000	FNNVUE	0	0	! listl (with no arguments)
T169	C7000	VSFVUE	0	0	vumcl ! listl
T172	C9020	FNNVUE	0	0	! wwdm n(ew)
T173	C9020	FNNVUE	0	0	! wwdms n(ew)
T174	C9020	FNNVUE	0	0	! wwdm new
T175	C9020	FNNVUE	0	0	! wwdms new

C200 T30	C999 C210	D30 FNNVSE	SNFN 0	LOGON 0	outnup ! Begin TLCF logon ! "TERMINAL "
C210 T40	C999 C220	D1 VSFVSE	SNVN 0	LOGON 0	outnup ! system identifies terminal termid ! "nn" (terminal id)
C220 T50 T51	C999 C230 C999	D50 FNNVSE FNFVSE	SNFN 0 0	LOGON 0 0	outnup ! system prompts for userid ! "USERID\$PASSWORD ... " locked ! "LOCKED" (terminal is locked)
C230 T60 T61	C240 C230 C260	D40 VSNIEU VSFVUE	UNVN 0 0	LOGON 0 0	outnup ! user enters userid ! <uid>\$<pw>/scc/scc [invalid] getuid ! <uid>\$<pw>
C240 T50	C999 C250	D40 FNNVSE	SRFN 0	LOGON 0	outnup ! system re-prompts for userid ! "USERID\$PASSWORD ... "
C250 T60 T61	C999 C250 C260	D40 VSNIEU VSFVUE	UNVN 0 0	LOGON 0 0	outnup ! user enters userid (last chan ! <uid>\$<pw>/scc/scc [invalid] getuid ! <uid>\$<pw>
C260 T90	C999 C270	D1 FNNVSE	SRFN 0	LOGON 0	outnup ! system prompts for ident ! "IDENT?"
C270 T100	C999 C280	D40 VSNVUE	UNVN 0	LOGON 0	outnup ! user enters identifier ! <ident>
C280 T110 T111 T90	C999 C290 C290 C270	D110 FNNVSE FNNVSE FNNVSE	SNFA 0 0 0	TLCF 0 0 0	outnup ! system responds to ident ! "CLASSIFICATION OF YOUR OUTPUT?" ! "classification of your output?" ! "IDENT?" (previous one bad)
C290 T121 T122 T123 T124 T125	C999 C11000 C11000 C11000 C11000 C11000	D40 FNFVUE FNFVUE FNFVUE FNFVUE FNFVUE	UNFN 0 0 0 0 0	TLCF 0 0 0 0 0	outnup ! user enters output class getcls ! uzz getcls ! ufo getcls ! czz getcls ! szz getcls ! swp
C990 T995 T991	C999 C999 C990	D990 VSNVSE FSNVSE	SNVN 0 0	TSS 0 0	exit ! system accounting info ! "***on at ... " ! "LINE TERMINATED CP"
C999 T1	C999 C999	D999 VNFVSE	SNVN 0	TSS 0	exit ! end of the line ! exit ! fsmgp termination !
C1000 T1000	C150 C1020	D1000 FNNVSE	SNFA 0	ACCESS 0	outnup ! begin access ! "FUNCTION?"
C1010	C150	D1000	SNFN	ACCESS	outnup ! system prompt for 2nd+ passes

T1000	C1020	FNNVSE	0	0	! "FUNCTION?"
C1020	C150	D40	UNFN	ACCESS	outnup ! user enters acecess function
T1020	C1030	FNNVUE	0	0	! mf (modify file)
T1021	C1030	FNNVUE	0	0	! fm (file modify)
T1022	C150	FNNVUE	0	0	! done
T1023	C150	FNNVUE	0	0	! <carriage return>
C1030	C150	D1030	SRFN	ACCESS	outnup ! prompt user for catalog
T1030	C1040	FNNVSE	0	0	! "CATALOG STRUCTURE TO WORKING LEVEL?"
C1040	C150	D40	UNVN	ACCESS	outnup ! user inputs catalog string
T1040	C1050	VNFVUE	0	0	vumc1 ! <cat string>
T1023	C1050	FNNVUE	0	0	! <carriage return> (only)
C1050	C150	D1050	SNFN	ACCESS	outnup ! system prompts for file
T1050	C1060	FNNVSE	0	0	! "FILE TO BE MODIFIED?"
T1051	C1030	VSINVSE	0	0	! "ERR-FIELD STARTING WITH...TOO LONG"
C1060	C150	D40	UNVN	ACCESS	outnup ! user enters file name
T1040	C1070	VNNVUE	0	0	! <filename>
T1023	C1010	FNNVUE	0	0	! <carriage return>
T1060	C1060	VSNIUE	0	0	! / [invalid]
C1070	C150	D1050	SNFN	ACCESS	outnup ! system prompts for new file n
T1070	C1080	FNNVSE	0	0	! "NEW NAME?"
T1051	C1050	VSINVSE	0	0	! "ERR-FIELD STARTING WITH...TOO LONG"
C1080	C150	D40	UNVN	ACCESS	outnup ! user enters new file name
T1040	C1090	VNNVUE	0	0	! <new filename>
C1090	C150	D1050	SNFN	ACCESS	outnup ! system prompts for file size
T1090	C1100	FNNVSE	0	0	! "NEW MAX SIZE IN LLINKS?"
T1051	C1070	VSINVSE	0	0	! "ERR-FIELD STARTING WITH...TOO LONG"
C1100	C150	D40	UNVN	ACCESS	outnup ! user enters file's new size
T1040	C1200	VNNVUE	0	0	! <new file size>
C1200	C150	D1200	SNFN	ACCESS	outnup ! system prompts for new passwo
T1200	C1210	FNNVSE	0	0	! "NEW PASSWORD?"
T1201	C1090	FNNVSE	0	0	! "ERR—MAX SIZE ILLEGALLY STATED"
C1210	C150	D40	UNVN	ACCESS	outnup ! user enters new password
T1040	C1220	VNNVUE	0	0	! <new password>
C1220	C150	D1050	SNVN	ACCESS	outnup ! system prompts for general pe
T1220	C1230	FNNVSE	0	0	! "GENERAL PERMISSIONS?"
T1051	C1200	VSINVSE	0	0	! "ERR-FIELD STARTING WITH...TOO LONG"
C1230	C150	D40	UNFN	ACCESS	outnup ! user enters general permissio

T1230	C1240	FNNVUE	0	0	! read
T1231	C1240	FNNVUE	0	0	! r (read)
T1232	C1240	FNNVUE	0	0	! execute
T1233	C1240	FNNVUE	0	0	! e (execute)
T1234	C1240	FNNVUE	0	0	! write
T1235	C1240	FNNVUE	0	0	! w (write)
T1236	C1240	FNNVUE	0	0	! append
T1237	C1240	FNNVUE	0	0	! a (append)
T1238	C1240	FNNVUE	0	0	! modify
T1239	C1240	FNNVUE	0	0	! m (modify)
T1023	C1240	FNNVUE	0	0	! <carriage return>
C1240	C150	D1240	SNVN	ACCESS	outnup ! system prompts user for speci
T1240	C1250	FNNVSE	0	0	! "SPECIFIC PERMISSIONS?"
T1051	C1220	VSNVSE	0	0	! "ERR-..." (several types)
C1250	C150	D40	UNVN	ACCESS	outnup ! user inputs specific permissi
T1250	C1300	VSNVUE	0	0	! read/
T1251	C1300	VSNVUE	0	0	! r/ (read)
T1252	C1300	VSNVUE	0	0	! write/
T1253	C1300	VSNVUE	0	0	! w/ (write)
T1254	C1300	VSNVUE	0	0	! append/
T1255	C1300	VSNVUE	0	0	! a/ (append)
T1256	C1300	VSNVUE	0	0	! execute/
T1257	C1300	VSNVUE	0	0	! e/ (execute)
T1258	C1300	VSNVUE	0	0	! modify/
T1259	C1300	VSNVUE	0	0	! m/ (modify)
T1023	C1300	FNNVUE	0	0	! <carriage return> (end specific permi
C1300	C150	D1300	SNFN	ACCESS	outnup ! response to above permissions
T1240	C1250	FNNVSE	0	0	! "SPECIFIC PERMISSIONS?"
T1300	C1310	FNNVSE	0	0	! "MORE?"
T1301	C1320	FNNVSE	0	0	! "SUCCESSFUL."
T1302	C1240	FNNVSE	0	0	! "ERR-PERMISSIONS ILLEGALLY STATED"
T1051	C1240	VSNVSE	0	0	! "ERR-..." (several types)
T1303	C1010	VSNVSE	0	0	! "REQUEST DENIED-"
C1310	C150	D40	UNVN	ACCESS	outnup ! user enters more names
T1040	C1300	VNNVUE	0	0	! (anything)
C1320	C150	D581	SNFN	ACCESS	outnup ! system prompts for file name
T1320	C1060	FNNVSE	0	0	! "FILE TO BE MODIFIED?"
C2000	C2004	D510	SNVA	TCON	outnup !tcon start1
T2002	C2010	FNNVSE	0	0	!<all>TCON VERSION 6.3, JUNE 1976<SP><<C
T2020	C150	VSNVSE	0	0	!NO TEMPORARY FILE SPACE CODE
C2001	C2004	D501	SNFN	TCON	outnup !tcon first question
T2001	C2010	FNNVSE	0	0	!<all>FUNCTIONAL AREA?

C2003	C2004	D40	UNFN	TCON	outnup !user hits <nl>
T2100	C2001	FNNVUE	0	0	!<all><CR>
C2004	C2004	D509	SNVN	TCON	outnup !back to system or top
T150	C160	FNNVSE	0	0	!SYSTEM ?
T150	C160	VSNVSE	0	0	!SYSTEM ? preceded by message
T2000	C2003	FNNVSE	0	0	!TCON VERSION 6.3, JUNE 1976
C2010	C2004	D40	UNFN	TCON	outnup !user input functional area
T2030	C2020	FNNVUE	0	0	!<all>fs
T2040	C4020	FNNVUE	0	0	!<all>dem
T2100	C150	FNNVUE	0	0	!<all><CR>
C2015	C2015	D509	SNVN	TCON	outnup !back to system, top or type?
T150	C160	FNNVSE	0	0	!SYSTEM ?
T150	C160	VSNVSE	0	0	!SYSTEM ? preceded by message
T2000	C2016	FNNVSE	0	0	!TCON VERSION 6.3, JUNE 1976
T2201	C2070	FNNVSE	0	0	!<CR><NL> TRANSACTION TYPE ?<SP><SP>
C2016	C2015	D40	UNFN	TCON	outnup !user hits <nl>
T2100	C2017	FNNVUE	0	0	!<CR>
C2017	C2015	D501	SNFN	TCON	outnup !tcon first question
T2001	C2018	FNNVSE	0	0	!FUNCTIONAL AREA?
C2018	C2015	D40	UNFN	TCON	outnup !user input functional area
T2030	C2020	FNNVUE	0	0	!fs
T2040	C4020	FNNVUE	0	0	!dem
T2100	C150	FNNVUE	0	0	!<CR>
C2020	C2015	D520	SNFN	TCON	outnup !system asks for old or new
T2110	C2030	FNNVSE	0	0	!<c2010(t2030)>OLD OR NEW-
T2110	C2030	VSNVSE	0	0	!OLD OR NEW-
T2190	C2020	FNNVSE	0	0	!<c2030(t2140)>READY
T2200	C2070	FNNVSE	0	0	!<all><CR><NL><NL><CR><NL>TRANSACTION T
T2201	C2070	FNNVSE	0	0	!<all><CR><NL> TRANSACTION TYPE ?<SP>
T2120	C150	FNNVSE	0	0	!FUNCTIONAL AREA BUSY
C2030	C2015	D40	UNVN	TCON	outnup !user response to old or new
T2130	C2020	FNNVUE	0	0	!<all>new<CR>
T2140	C2020	FNNVUE	0	0	!<all>n<CR>
T2640	C3000	FNNVUE	0	0	!<all>old<CR>
T2645	C3000	FNNVUE	0	0	!<all>o<CR>
T2162	C2800	VS FVUE	0	0	vumc2 !<all>old <filename>
T2160	C2800	VSFVUE	0	0	vumc2 !<all>o <filename>
T2180	C2042	FNNVUE	0	0	!<all>same<CR>
T2181	C2042	FNNVUE	0	0	!<all>s<CR>
T2100	C150	FNNVUE	0	0	!<all><CR>
C2042	C2015	D530	SNFN	TCON	outnup !sys resp to same

T2210	C2020	FNNVSE	0	0	!<all>YOU PRESENTLY DO NOT HAVE A CURRE
T2190	C2800	FNNVSE	0	0	!READY
C2050	C2015	D510	SNVN	TCON	outnup !tcon start2
T2002	C2018	FNNVSE	0	0	!TCON VERSION 6.3, JUNE 1976
T2020	C150	VSNVSE	0	0	!NO TEMPORARY FILE SPACE CODE
C2070	C2015	D40	UNVN	TCON	outnup !user input transaction type
T2290	C2692	FNNVUE	0	0	!<c2020(t2200)>a
T2300	C2692	FNNVUE	0	0	!<c2020(t2200)>a a
T2310	C2692	FNNVUE	0	0	!<c2020(t2200)>aa1
T2320	C2692	FNNVUE	0	0	!<c2020(t2200)>d
T2330	C2692	FNNVUE	0	0	!<c2020(t2200)>e
T2340	C2692	FNNVUE	0	0	!<c2020(t2200)>h
T2350	C2692	FNNVUE	0	0	!j
T2360	C2692	FNNVUE	0	0	!k
T2370	C2692	FNNVUE	0	0	!ka1
T2380	C2692	FNNVUE	0	0	!l
T2390	C2692	FNNVUE	0	0	!m
T2400	C2692	FNNVUE	0	0	!n
T2410	C2692	FNNVUE	0	0	!p
T2420	C2692	FNNVUE	0	0	!r-1
T2430	C2692	FNNVUE	0	0	!r-2
T2440	C2692	FNNVUE	0	0	!r11
T2450	C2692	FNNVUE	0	0	!r12
T2460	C2692	FNNVUE	0	0	!aa4
T2470	C2692	FNNVUE	0	0	!ka2
T2480	C2692	FNNVUE	0	0	!t
T2490	C2692	FNNVUE	0	0	!v
T2500	C2692	FNNVUE	0	0	!x
T2510	C2692	FNNVUE	0	0	!z a
T2512	C2692	FNNVUE	0	0	!jsb
T2514	C2692	FNNVUE	0	0	!j b
T2516	C2692	FNNVUE	0	0	!n a
T2518	C2692	FNNVUE	0	0	!n b
T2519	C2692	FNNVUE	0	0	!xxx
T3030	C990	FNNVUE	0	0	!<c2040>bye
T2100	C2050	FNNVUE	0	0	!<c2040 c2072 c2020(t2201)><CR>
T2540	C2072	VSNVUE	0	0	!<c2040 c2072>dele(te) <transaction>
T2520	C150	FNNVUE	0	0	!<c2040>done<CR>
T2610	C2071	FNNVUE	0	0	!<c2040>list<CR>
T2612	C2071	FNNVUE	0	0	!listl<CR>
T2611	C2071	VS FVUE	0	0	vumcl !<c2040>list <filename>
T2613	C2071	VS FVUE	0	0	vumcl !listl <filename>
T2640	C3000	FNNVUE	0	0	!<c2072(t2202)>old<CR>
T2162	C2800	VS FVUE	0	0	vumc2 !old <filename>
T2550	C2072	FNNVUE	0	0	!<c2072>remove clearfiles
T2555	C2072	FNNVUE	0	0	!<c2072 c2040>remov clearfiles
T2560	C2072	FNNVUE	0	0	!<c2040 c2072>remo clearfiles
T2630	C2075	FNNVUE	0	0	!<c2072(t2202)>resave<CR>

T2631	C2075	FNNVUE	0	0	!<c2072(t2202)>resa<CR>
T2632	C2077	VSVFVUE	0	0	vumc2 !<c2077>resave <filename>
T2633	C2077	VSVFVUE	0	0	vumc2 !<c2077>resa <filename>
T2580	C2072	VSNVUE	0	0	!<c2074>rese(quence)
T2600	C2074	VSNVUE	0	0	!<c2020(t2200) c2072(t2202)>retr
C2071	C2015	D506	SNVN	TCON	outnup !sys resp to list
T2280	C2072	FNNVSE	0	0	!<all>ready
T2281	C2072	VSNVSE	0	0	!ready preceded by data/error
C2072	C2072	D1000	SNFN	TCON	outnup !sys resp after ready(list), re
T2201	C2070	FNNVSE	0	0	!<c2070(t2540)><CR><NL> TRANSACTION
T2202	C2070	FNNVSE	0	0	!<c2060 c2070(t2560,t2555,t2550,t2580)
C2074	C2072	D599	SNVN	TCON	outnup !sys resp after retr
T2250	C2072	FNNVSE	0	0	!<CR><NL>ILLEGAL PARAMETER<SP><SP><SP>
T2202	C2070	VSNVSE	0	0	!<all><CR><NL>TRANSACTION TYPE ?<SP><SP>
C2075	C2072	D581	SNFN	TCON	outnup !file name for resave
T2211	C2076	FNNVSE	0	0	!<all><CR><NL>FILE NAME?<SP>
C2076	C2072	D40	UNVN	TCON	outnup !user input file name
T2100	C2072	FNNVUE	0	0	!<all><CR>
T2100	C2077	VSVFVUE	0	0	vumc1 !<all><filename>
C2077	C2072	D502	SNVN	TCON	outnup !sys resp to file name
T2221	C2072	FNNVSE	0	0	!<c2070(t2633)><CR><NL><NL>ERR-FILE NAM
T2202	C2070	VSNVSE	0	0	!<c2076 c2070(t2632)><CR><NL>TRANSACTION
C2080	C2691	D560	SRFN	TCON	outnup !a skeleton
T2660	C2690	FNNVSE	0	0	!<all>a skeleton
C2100	C2691	D560	SRFN	TCON	outnup !a a skeleton
T2670	C2690	FNNVSE	0	0	!<all>a a skeleton
C2120	C2691	D560	SRFN	TCON	outnup !aa1 skeleton
T2680	C2690	FNNVSE	0	0	!aa1 skeleton
C2140	C2691	D560	SRFN	TCON	outnup !d skeleton
T2690	C2690	FNNVSE	0	0	!<all>d skeleton
C2160	C2691	D560	SRFN	TCON	outnup !e skeleton
T2700	C2690	FNNVSE	0	0	!<all>e skeleton
C2180	C2691	D560	SRFN	TCON	outnup !h skeleton
T2710	C2690	FNNVSE	0	0	!<all>h skeleton
C2200	C2691	D560	SRFN	TCON	outnup !j skeleton
T2720	C2690	FNNVSE	0	0	!<all>skeleton



C2220 T2730	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !k skeleton !<all>skeleton
C2240 T2740	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !ka1 skeleton !<all>skeleton
C2260 T2750	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !l skeleton !<all>skeleton
C2280 T2760	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !m skeleton !<all>skeleton
C2300 T2770	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !n skeleton !<all>skeleton
C2320 T2780	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !p skeleton !<all>skeleton
C2340 T2790	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !r-1 skeleton !<all>skeleton
C2360 T2800	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !r-2 skeleton !skeleton
C2380 T2810	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !r11 skeleton !skeleton
C2400 T2820	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !r12 skeleton !skeleton
C2420 T2830	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !aa4 skeleton !skeleton
C2440 T2840	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !ka2 skeleton !skeleton
C2460 T2850	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !t skeleton !skeleton
C2480 T2860	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !v skeleton !skeleton
C2500 T2870	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !x skeleton !skeleton
C2520 T2880	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !z a skeleton !skeleton
C2540 T3140	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !jsb skeleton !skeleton

C2560 T3150	C2691 C2690	D560 FNNVSE	SRFN U	TCON 0	outnup !j b skeleton !skeleton
C2580 T3160	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !n a skeleton !skeleton
C2600 T3170	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !n b skeleton !skeleton
C2620 T3180	C2691 C2690	D560 FNNVSE	SRFN 0	TCON 0	outnup !xxx skeleton !skeleton
C2690 T3070	C2691 C2700	D550 FNNVSE	SNFN 0	TCON 0	outnup !end of skeletons !<c2080c2100c2120c2140c2160 2180c2200c2
C2691 T2050	C2710 C2691	D560 VSNVSE	SRVN 0	TCON 0	outnup !remnants of skeleton after <br !<CR><NL>
C2692 T3070	C2710 C2700	D560 VSNVSE	SRVN 0	TCON 0	outnup !variable skeleton !<c2070(t2290t2300t2310t2320t2330t2340)
C2700 T3030	C2710 C990	D40 FNNVUE	UNVN 0	TCON 0	outnup !user skeleton data entry !<c2710(t2050)c2720(t3070)>bye
T3040	C2800	FNNVUE	0	0	!<c2690 c2710(t2050)c3071>d (done with
T2520	C150	FNNVUE	0	0	!<c2690>done<CR>
T2610	C2730	FNNVUE	0	0	!<c2710(t2050)c3071>list<CR>
T2612	C2730	FNNVUE	0	0	!<c3071(t3070)>listl<CR>
T2640	C3000	FNNVUE	0	0	!<c2692c2710(t2050)>old<CR>
T2550	C2710	FNNVUE	0	0	!<c2692c2721(t3070)>remove clearfiles
T2555	C2710	FNNVUE	0	0	!<c2710(t2050)>remov clearfiles
T2560	C2710	FNNVUE	0	0	!<c2710(t2050)>remo clearfiles
T2630	C2711	FNNVUE	0	0	!resave<CR>
T2631	C2711	FNNVUE	0	0	!resa<CR>
T3011	C2700	FNNIUE	0	0	!<c2710(t2050)>system<CR>
T3012	C2700	FNNIUE	0	0	!<c2701(t2050)>syst<CR>
T3000	C2720	FNNVUE	0	0	!<c2710(t2050)>test<CR>
T3001	C2720	FNNVUE	0	0	!<c2710(t2050)>t<CR>
T2890	C2720	FNNVUE	0	0	!<c2690>tn<CR>
T3020	C2720	FNNVUE	0	0	!<c2720(t3070)c2732>xmit<CR>
T3021	C2720	FNNVUE	0	0	!<c2710(t2050)>x<CR>
T3010	C2710	FNNVUE	0	0	!<c2690 c2720(t3070)c3071>*<CR>
T2100	C2720	FNNVUE	0	0	!<c2690 c2710(t2050)><CR>
T2960	C2720	VSNVUE	0	0	!<c2710(t2050)c2690>backward
T2540	C2800	VSNVUE	0	0	!<c2692c3071>dele(te)
T2541	C2700	VSNIUE	0	0	!<all>feed(back)
T2940	C2720	VSNVUE	0	0	!<c2721(t3080)>forward
T2980	C2720	VSNVUE	0	0	!<c2710(t2050)c2721(t3070)>get<SP>
T2547	C2700	VSNIUE	0	0	!<all>help
T2611	C2730	VSFVUE	0	0	vumcl !<c2720(t3070)>list <filename>
T2613	C2730	VSFVUE	0	0	vumcl !<c2710(t2050)>listl <filename>

T2542	C2700	VSNIUE	0	0	!<all>next
T2543	C2700	VSNIUE	0	0	!<all>nofe(ed)
T2162	C2800	VSFVUE	0	0	vumc2 !<c2692c2721(t3080)c3071>old <f
T2544	C2700	VSNIUE	0	0	!<all>pass
T2545	C2700	VSNIUE	0	0	!<all>rewi(nd)
T2546	C2700	VSNIUE	0	0	!<all>skip
T2548	C2700	VSNIUE	0	0	!<all>voca(bulary)
T2632	C2713	VSFVUE	0	0	vumc2 !resave <filename>
T2633	C2713	VSFVUE	0	0	vumc2 !resa <filename>
T2580	C2710	VSNVUE	0	0	!<all>rese(quence)
T2600	C2731	VSNVUE	0	0	!<c2692c2710(t2050)c2721(t3080)c3071>re
T2161	C2710	VSNVUE	0	0	!<c2690>ol <filename>
T2981	C2720	VSNVUE	0	0	!<c2721(3070)>g<SP>
T2160	C2710	VSNVUE	0	0	!<c2690>o <filename>
T2891	C2710	VSNVUE	0	0	!<c2690>tn
T2910	C2710	VSNVUE	0	0	!<c2720(t3070)>[
T2060	C2710	VSNVUE	0	0	!0<SP>
T2061	C2710	VSNVUE	0	0	!1<SP>
T2062	C2710	VSNVUE	0	0	!2<SP>
T2063	C2710	VSNVUE	0	0	!3<SP>
T2064	C2710	VSNVUE	0	0	!4<SP>
T2065	C2710	VSNVUE	0	0	!5<SP>
T2066	C2710	VSNVUE	0	0	!6<SP>
T2067	C2710	VSNVUE	0	0	!7<SP>
T2068	C2710	VSNVUE	0	0	!8<SP>
T2069	C2710	VSNVUE	0	0	!9<SP>
T2961	C2720	VSNVUE	0	0	!<c2721(t3070)>[
T2941	C2720	VSNVUE	0	0	!<c2721(t3070)>f
C2710	C2710	D570	SNVN	TCON	outnup !sys resp to skel data entry
T2050	C2700	FNNVSE	0	0	!<c2800c2730(t2281)c2731(t2240t2250)c26
T3060	C2710	FNNVSE	0	0	!<c2700(t2160t3010)><CR><NL>INVALID COM
T3050	C2715	VSNVSE	0	0	! DATA TRUNCATED<SP>
C2711	C2710	D581	SNFN	TCON	outnup !sys resp to resave
T2211	C2712	FNNVSE	0	0	!<c2700(t2631)>FILE NAME?<SP>
C2712	C2800	D40	UNVN	TCON	outnup !user input filename
T2100	C2710	FNNVUE	0	0	!<all><CR>
T2100	C2713	VNFVUE	0	0	vumc1 !<all><filename>
C2713	C2710	D539	SNVN	TCON	outnup !sys resp to filename
T2270	C2714	FNNVSE	0	0	!<CR><NL>DATA SAVED-
T2196	C2710	VSNVSE	0	0	!FILE '<filename>' DOES NOT EXIST--USE
T2220	C2710	VSNVSE	0	0	!<CR><NL><50>FILE <filename> -- INCORRE
T2221	C2710	FNNVSE	0	0	!<CR><NL><NL>ERR-FILE NAME >8 CHARACTER
T2224	C2710	VSNVSE	0	0	!<CR><NL><NL>ERR-'<character>' IS AN IL
T2225	C2710	VSNVSE	0	0	!<CR><NL><NL>ERR-FIELD STARTING WITH '<
C2714	C2710	D560	SNVN	TCON	outnup !end of data saved msg

T2100	C2700	VNNVSE	0	0	!<filename><CR><NL>
C2715	C2710	D40	UNVN	TCON	outnup !  cr ^
T2100	C2710	VSNVUE	0	0	!anything
C2720	C2710	D580	SNVN	TCON	outnup ! ? ^E cr  nl ^R  cr  nl ^LEVE
T2240	C2710	VSNVSE	0	0	!TRANSACTION xxxxxx NOT IN CURR FILE<SP
T2657	C2721	FNNVSE	0	0	!<c2700(t2980t2981)><CR><NL>spaces<CR><
T2658	C2721	VSNVSE	0	0	!TRANSACTION SEQUENCE NUMBER <CR><NL>
T2659	C2721	VSNVSE	0	0	!<c2700(t2960t2961t2940t2941)>ERROR MES
T3090	C3060	FNNVSE	0	0	!<c2700(t3020)>TRANSACTION SENT. NEXT?<
T3090	C3060	VSNVSE	0	0	!TRANSACTION SENT. NEXT?<SP>
T3060	C2710	FNNVSE	0	0	!<CR><NL>INVALID COMMAND AT THIS LEVEL<
C2721	C2710	D560	SRVN	TCON	outnup !  cr  nl ^
T3070	C2700	VSNVSE	0	0	!<c2720(t2657t2659)>END OF TRANSACTION
T3080	C2700	VSNVSE	0	0	!<c2720(t2659t2657)>MORE - - - -<CR><NL
C2730	C2800	D506	SNVN	TCON	outnup !sys out of list
T2280	C2710	FNNVSE	0	0	!ready
T2281	C2710	VSNVSE	0	0	!<c2700(t2610t2611t2612t2613)>ready pre
C2731	C2800	D598	SNVN	TCON	outnup !  cr  nl ^TYPE ^- ^
T2205	C2733	FNNVSE	0	0	!<c2700(t2600)>INVALID TRANSACTION TYPE
T2240	C2710	VSNVSE	0	0	!<c2700(t2600)>TRANSACTION xxxxxx NOT I
T2250	C2710	FNNVSE	0	0	!<c2700(t2600)><CR><NL>ILLEGAL PARAMETE
T2657	C2732	FNNVSE	0	0	!<CR><NL>spaces<CR><NL>TN[000010spacest
T2658	C2732	VSNVSE	0	0	!<c2700(t2600)>TRANSACTION SEQUENCE NUM
C2732	C2800	D560	SRVN	TCON	outnup ! <CR><NL>
T3070	C2700	VSNVSE	0	0	!<c2731(t2658)>END OF TRANSACTION <CR>
C2733	C2731	D40	UNFN	TCON	!user enters tran type
T2290	C3070	FNNVUE	0	0	!<c2731>a
T2300	C3070	FNNVUE	0	0	!<c2731>a a
T2310	C3070	FNNVUE	0	0	!aa1
T2320	C3070	FNNVUE	0	0	!d
T2330	C3070	FNNVUE	0	0	!e
T2340	C3070	FNNVUE	0	0	!h
T2350	C3070	FNNVUE	0	0	!j
T2360	C3070	FNNVUE	0	0	!k
T2370	C3070	FNNVUE	0	0	!ka1
T2380	C3070	FNNVUE	0	0	!l
T2390	C3070	FNNVUE	0	0	!m
T2400	C3070	FNNVUE	0	0	!n
T2410	C3070	FNNVUE	0	0	!p
T2420	C3070	FNNVUE	0	0	!r-1
T2430	C3070	FNNVUE	0	0	!r-2
T2440	C3070	FNNVUE	0	0	!r11
T2450	C3070	FNNVUE	0	0	!r12

T2460	C3070	FNNVUE	0	0	!aa4
T2470	C3070	FNNVUE	0	0	!ka2
T2480	C3070	FNNVUE	0	0	!t
T2490	C3070	FNNVUE	0	0	!v
T2500	C3070	FNNVUE	0	0	!x
T2510	C3070	FNNVUE	0	0	!z a
T2512	C3070	FNNVUE	0	0	!jsb
T2514	C3070	FNNVUE	0	0	!j b
T2516	C3070	FNNVUE	0	0	!n a
T2518	C3070	FNNVUE	0	0	!n b
T2519	C3070	FNNVUE	0	0	!xxx
C2800	C4004	D540	SNVN	TCON	outnup !variable tran type
T2110	C2990	FNNVSE	0	0	!OLD OR NEW-
T2110	C2990	VSNVSE	0	0	!OLD OR NEW-
T2190	C2800	FNNVSE	0	0	!<c2700(t2162)c3010c3060(t2162)>READY
T2200	C3060	FNNVSE	0	0	!<c2800><CR><NL><NL><CR><NL>TRANSACTION
T2200	C3060	VSNVSE	0	0	!<CR><NL><NL><CR><NL>TRANSACTION TYPE ?
T2201	C3060	FNNVSE	0	0	!<c2700(t2540t3040) c2992 c2800(t2190)c
T2201	C3060	VSNVSE	0	0	!<CR><NL> TRANSACTION TYPE ?<SP><SP>
T2222	C2800	FNNVSE	0	0	!<CR><NL>ERR-FILE NAME >8 CHARACTERS
C2990	C4004	D40	UNVN	TCON	outnup !user response to old or new
T2130	C2020	FNNVUE	0	0	!new<CR>
T2140	C2020	FNNVUE	0	0	!n<CR>
T2640	C3000	FNNVUE	0	0	!old<CR>
T2645	C3000	FNNVUE	0	0	!o<CR>
T2162	C2800	VS FVUE	0	0	vumc2 !old <filename>
T2160	C2800	VS FVUE	0	0	vumc2 !o <filename>
T2180	C2992	FNNVUE	0	0	!same<CR>
T2181	C2992	FNNVUE	0	0	!s<CR>
T2100	C150	FNNVUE	0	0	!<CR>
C2992	C4004	D530	SNFN	TCON	outnup !sys resp to same
T2210	C2800	FNNVSE	0	0	!YOU PRESENTLY DO NOT HAVE A CURRENT FI
T2190	C2800	FNNVSE	0	0	!<c2990(t2180)>READY
C3000	C2800	D581	SNFN	TCON	outnup !old file path
T3110	C3010	FNNVSE	0	0	!<c2030 c2070(t2640) c2700(t2640) c3060
C3010	C2800	D40	UNVN	TCON	outnup !user input old filename
T2100	C2800	FNNVUE	0	0	!<CR>
T2100	C2800	VNFVUE	0	0	vumc1 !<filename>
C3060	C2710	D40	UNVN	TCON	outnup !user input transaction type
T2290	C3070	FNNVUE	0	0	!<c2800(t2200t2201)c3062(t2202)>a
T2300	C3070	FNNVUE	0	0	!<c2800(t2201)>a a
T2310	C3070	FNNVUE	0	0	!<c2800(t2201)>aa1
T2320	C3070	FNNVUE	0	0	!<c2800(t2201)>d
T2330	C3070	FNNVUE	0	0	!<c2800(t2201)>e

T2340	C3070	FNNVUE	0	0	!<c2800(t2201)>>h
T2350	C3070	FNNVUE	0	0	!<c2800(t2201)>>j
T2360	C3070	FNNVUE	0	0	!k
T2370	C3070	FNNVUE	0	0	!ka1
T2380	C3070	FNNVUE	0	0	!l
T2390	C3070	FNNVUE	0	0	!m
T2400	C3070	FNNVUE	0	0	!n
T2410	C3070	FNNVUE	0	0	!p
T2420	C3070	FNNVUE	0	0	!r-1
T2430	C3070	FNNVUE	0	0	!r-2
T2440	C3070	FNNVUE	0	0	!r11
T2450	C3070	FNNVUE	0	0	!r12
T2460	C3070	FNNVUE	0	0	!aa4
T2470	C3070	FNNVUE	0	0	!ka2
T2480	C3070	FNNVUE	0	0	!t
T2490	C3070	FNNVUE	0	0	!v
T2500	C3070	FNNVUE	0	0	!x
T2510	C3070	FNNVUE	0	0	!z a
T2512	C3070	FNNVUE	0	0	!jsb
T2514	C3070	FNNVUE	0	0	!j b
T2516	C3070	FNNVUE	0	0	!n a
T2518	C3070	FNNVUE	0	0	!n b
T2519	C3070	FNNVUE	0	0	!xxx
T3010	C3070	FNNVUE	0	0	!*
T3030	C990	FNNVUE	0	0	!<c2042 c3062(t2201t2202)>>bye
T2100	C4000	FNNVUE	0	0	!<c2800(t2201)c3062(t2202)>><CR>
T2540	C3062	VSNVUE	0	0	!<c3062(t2202)>>dele(te) <transaction>
T2520	C150	FNNVUE	0	0	!<c2720c2800(t2201)>>done<CR>
T2610	C3061	FNNVUE	0	0	!<c2800(t2201)>>list<CR>
T2612	C3061	FNNVUE	0	0	!<c2800(t2201)>>listl<CR>
T2611	C3061	VS FVUE	0	0	vumcl !<c2800(t2201)>>list <filename>
T2613	C3061	VS FVUE	0	0	vumcl !<c3062(t2202)>>listl <filename>
T2640	C3000	FNNVUE	0	0	!<c3062(t2202)>>old<CR>
T2162	C2800	VS FVUE	0	0	vumc2 !<c3067>>old <filename>
T2550	C3062	FNNVUE	0	0	!<c3062(t2201)>>remove clearfiles
T2555	C3062	FNNVUE	0	0	!<c3062(t2202)>>remov clearfiles
T2560	C3062	FNNVUE	0	0	!<c3062(t2202)>>remo clearfiles
T2630	C3063	FNNVUE	0	0	!<c2720>>resave<CR>
T2631	C3063	FNNVUE	0	0	!<c3062(t2202)>>resa<CR>
T2632	C3065	VS FVUE	0	0	vumc2 !resave <filename>
T2633	C3065	VS FVUE	0	0	vumc2 !resa <filename>
T2580	C3062	VSNVUE	0	0	!<c3062(t2202)>>rese(quence)
T2600	C3070	VSNVUE	0	0	!<c2800(t2200t2201)c3062(t2202)>>retr
C3061	C4004	D506	SNVN	TCON	outnup !sys resp to list
T2280	C3062	FNNVSE	0	0	!<c3060(t2610)>>ready
T2281	C3062	VSNVSE	0	0	!<c3060(t2610t2612 t2613)>>ready precede
C3062	C3062	D1000	SNFN	TCON	outnup !sys resp after ready(list), re
T2201	C3060	FNNVSE	0	0	!<c3060(t2540)c3070(pop)>><CR><NL> TR

T2202	C3060	FNNVSE	0	0	!<c3060(t2550t2555t2560t2580)c3061c3064
C3063	C3062	D581	SNFN	TCON	outnup !sys resp to resave
T2211	C3064	FNNVSE	0	0	!<all>FILE NAME?<SP>
C3064	C3062	D40	UNVN	TCON	outnup !user input filename
T2100	C3062	FNNVUE	0	0	!<all><CR>
T2100	C3065	VNFVUE	0	0	vumc1 !<all><filename>
C3065	C3062	D502	SNVN	TCON	outnup !sys resp to resave
T2202	C3060	VSNVSE	0	0	!<CR><NL>TRANSACTION TYPE ?<SP><SP> pre
T2221	C3062	FNNVSE	0	0	!<CR><NL><NL>ERR-FILE NAME >8 CHARACTER
C3070	C3062	D598	SNVN	TCON	outnup !<t2205t2230>  cr  nl ^CURR FIL
T2230	C3060	FNNVSE	0	0	!INVALID TRANSACTION TYPE -RETYPE-<SP><
T2205	C2733	FNNVSE	0	0	!INVALID TRANSACTION TYPE ON RETRIEVE<C
T2240	C3062	VSNVSE	0	0	!TRANSACTION xxxxxx NOT IN CURR FILE<SP
T2250	C3062	FNNVSE	0	0	!<c3060(t2600)>><CR><NL>ILLEGAL PARAMETE
T2657	C3071	FNNVSE	0	0	!<c3060(t2300t2310)>><CR><NL>spaces<CR><
T2658	C3071	VSNVSE	0	0	!<c2733(t2290t2300)c3060(t2290t2320t234
C3071	C2691	D560	SRVN	TCON	outnup !  cr  nl ^
T3070	C2700	VSNVSE	0	0	!<c3070(t2657t2658)>>END OF TRANSACTION
C4000	C4004	D510	SNVN	TCON	outnup !tcon start3
T2002	C4010	FNNVSE	0	0	!<all>TCON VERSION 6.3, JUNE 1976
T2020	C150	VSNVSE	0	0	!NO TEMPORARY FILE SPACE CODE
C4001	C4004	D501	SNFN	TCON	outnup !tcon first question
T2001	C4010	FNNVSE	0	0	!<all>FUNCTIONAL AREA?
C4003	C4004	D40	UNFN	TCON	outnup !user hits <nl>
T2100	C4001	FNNVUE	0	0	!<all><CR>
C4004	C4004	D509	SNVN	TCON	outnup !back to system, top or type?
T150	C160	FNNVSE	0	0	!SYSTEM ?
T150	C160	VSNVSE	0	0	!SYSTEM ? preceded by message
T2000	C4003	FNNVSE	0	0	!<c2800>TCON VERSION 6.3, JUNE 1976
T2201	C3060	FNNVSE	0	0	!<CR><NL> TRANSACTION TYPE ?<SP><SP>
C4010	C4004	D40	UNFN	TCON	outnup !user input functional area
T2030	C2800	FNNVUE	0	0	!<all>fs
T2040	C4800	FNNVUE	0	0	!<all>dem
T2100	C150	FNNVUE	0	0	!<all><CR>
C4015	C4015	D509	SNVN	TCON	outnup !back to system, top or type?
T150	C160	FNNVSE	0	0	!SYSTEM ?
T150	C160	VSNVSE	0	0	!SYSTEM ? preceded by message
T2000	C4016	FNNVSE	0	0	!TCON VERSION 6.3, JUNE 1976
T2201	C4070	FNNVSE	0	0	!<CR><NL> TRANSACTION TYPE ?<SP><SP>

C4016	C4015	D40	UNFN	TCON	outnup !user hits <nl>
T2100	C4017	FNNVUE	0	0	!<CR>
C4017	C4015	D501	SNFN	TCON	outnup !tcon first question
T2001	C4018	FNNVSE	0	0	!FUNCTIONAL AREA?
C4018	C4015	D40	UNFN	TCON	outnup !user input functional area
T2030	C4020	FNNVUE	0	0	!fs
T2040	C4020	FNNVUE	0	0	!dem
T2100	C150	FNNVUE	0	0	!<CR>
C4020	C4015	D520	SNFN	TCON	outnup !system asks for old or new
T2110	C4030	FNNVSE	0	0	!<c2010>OLD OR NEW-
T2110	C4030	VSNVSE	0	0	!OLD OR NEW-
T2190	C4020	FNNVSE	0	0	!<c4030(t2130t2140)>READY
T2200	C4070	FNNVSE	0	0	!<c4020(t2190)><CR><NL><NL><CR><NL>TRAN
T2201	C4070	FNNVSE	0	0	!<CR><NL> TRANSACTION TYPE ?<SP><SP>
T2120	C150	FNNVSE	0	0	!FUNCTIONAL AREA BUSY
C4030	C4015	D40	UNVN	TCON	outnup !user response to old or new
T2130	C4020	FNNVUE	0	0	!<all>new<CR>
T2140	C4020	FNNVUE	0	0	!<all>n<CR>
T2640	C5020	FNNVUE	0	0	!<all>old<CR>
T2645	C5020	FNNVUE	0	0	!<all>o<CR>
T2162	C4800	VS FVUE	0	0	vumc2 !old <filename>
T2160	C4800	VS FVUE	0	0	vumc2 !o <filename>
T2180	C4042	FNNVUE	0	0	!same<CR>
T2181	C4042	FNNVUE	0	0	!<all>s<CR>
T2100	C150	FNNVUE	0	0	!<CR>
C4042	C4015	D530	SNFN	TCON	outnup !sys.resp to same
T2210	C4020	FNNVSE	0	0	!YOU PRESENTLY DO NOT HAVE A CURRENT FI
T2190	C4800	FNNVSE	0	0	!READY
C4050	C4015	D510	SNVN	TCON	outnup !tcon start4
T2002	C4018	FNNVSE	0	0	!TCON VERSION 6.3, JUNE 1976
T2020	C150	VSNVSE	0	0	!NO TEMPORARY FILE SPACE CODE
C4070	C4015	D40	UNVN	TCON	outnup !user input tran type
T3300	C4342	FNNVUE	0	0	!<c4020(t2200)>hd
T3310	C4342	FNNVUE	0	0	!<c4020(t2200)>as
T3320	C4342	FNNVUE	0	0	!am
T3330	C4342	FNNVUE	0	0	!av
T3340	C4342	FNNVUE	0	0	!ae
T3350	C4342	FNNVUE	0	0	!at
T3380	C4342	FNNVUE	0	0	!ma
T3390	C4342	FNNVUE	0	0	!mc
T3400	C4342	FNNVUE	0	0	!pl
T3410	C4342	FNNVUE	0	0	!pd
T3420	C4342	FNNVUE	0	0	!ol



T3430	C4342	FNNVUE	0	0	!od
T3030	C990	FNNVUE	0	0	!bye
T2100	C4050	FNNVUE	0	0	!<CR>
T2520	C150	FNNVUE	0	0	!<c4020(t2200)>done<CR>
T2540	C4072	VSNVUE	0	0	!dele(te) <transaction>
T2610	C4071	FNNVUE	0	0	!list<CR>
T2612	C4071	FNNVUE	0	0	!listl<CR>
T2611	C4071	VSFVUE	0	0	vumcl !list <filename>
T2613	C4071	VSFVUE	0	0	vumcl !listl <filename>
T2640	C5020	FNNVUE	0	0	!old<CR>
T2162	C4800	VSFVUE	0	0	vumc2 !old <filename>
T2550	C4072	FNNVUE	0	0	!remove clearfiles
T2555	C4072	FNNVUE	0	0	!remov clearfiles
T2560	C4072	FNNVUE	0	0	!remo clearfiles
T2630	C4075	FNNVUE	0	0	!resave<CR>
T2631	C4075	FNNVUE	0	0	!resa<CR>
T2632	C4077	VSFVUE	0	0	vumc2 !resave <filename>
T2633	C4077	VSFVUE	0	0	vumc2 !resa <filename>
T2580	C4072	VSNVUE	0	0	!rese(quence)
T2600	C4074	VSNVUE	0	0	!retr
C4071	C4015	D506	SNVN	TCON	outnup !sys resp to list
T2280	C4072	FNNVSE	0	0	!ready
T2281	C4072	VSNVSE	0	0	!ready preceded by data/error
C4072	C4072	D1000	SNFN	TCON	outnup !sys resp after ready(list), re
T2201	C4070	FNNVSE	0	0	!<CR><NL> TRANSACTION TYPE ?<SP><SP>
T2202	C4070	FNNVSE	0	0	!<CR><NL>TRANSACTION TYPE ?<SP><SP>
C4074	C4072	D599	SNVN	TCON	outnup !sys resp after retr
T2250	C4072	FNNVSE	0	0	!<CR><NL>ILLEGAL PARAMETER<SP><SP><SP>
T2202	C4070	VSNVSE	0	0	!<CR><NL>TRANSACTION TYPE ?<SP><SP>prec
C4075	C4072	D581	SNFN	TCON	outnup !file name for resave
T2211	C4076	FNNVSE	0	0	!<CR><NL>FILE NAME?<SP>
C4076	C4072	D40	UNVN	TCON	outnup !user input file name
T2100	C4072	FNNVUE	0	0	!<CR>
T2100	C4077	VSFVUE	0	0	vumc1 !<filename>
C4077	C4072	D502	SNVN	TCON	outnup !sys resp to file name
T2221	C4072	FNNVSE	0	0	!<CR><NL><NL>ERR-FILE NAME >8 CHARACTER
T2202	C4070	VSNVSE	0	0	!<CR><NL>TRANSACTION TYPE ?<SP><SP>prec
C4080	C4341	D560	SRFN	TCON	outnup !hd skeleton
T3440	C4340	FNNVSE	0	0	!skeleton
C4085	C4341	D560	SRFN	TCON	outnup !as skeleton
T3450	C4340	FNNVSE	0	0	!skeleton

C4090	C4341	D560	SRFN	TCON	outnup !am skeleton
T3460	C4340	FNNVSE	0	0	!skeleton
C4110	C4341	D560	SRFN	TCON	outnup !av skeleton
T3470	C4340	FNNVSE	0	0	!skeleton
C4130	C4341	D560	SRFN	TCON	outnup !ae skeleton
T3480	C4340	FNNVSE	0	0	!skeleton
C4150	C4341	D560	SRFN	TCON	outnup !at skeleton
T3490	C4340	FNNVSE	0	0	!skeleton
C4210	C4341	D560	SRFN	TCON	outnup !ma skeleton
T3520	C4340	FNNVSE	0	0	!skeleton
C4230	C4341	D560	SRFN	TCON	outnup !mc skeleton
T3530	C4340	FNNVSE	0	0	!skeleton
C4250	C4341	D560	SRFN	TCON	outnup !pl skeleton
T3540	C4340	FNNVSE	0	0	!skeleton
C4270	C4341	D560	SRFN	TCON	outnup !pd skeleton
T3550	C4340	FNNVSE	0	0	!skeleton
C4290	C4341	D560	SRFN	TCON	outnup !ol skeleton
T3560	C4340	FNNVSE	0	0	!skeleton
C4310	C4341	D560	SRFN	TCON	outnup !od skeleton
T3570	C4340	FNNVSE	0	0	!skeleton
C4340	C4341	D550	SNFN	TCON	outnup !end of skeletons
T3070	C4700	FNNVSE	0	0	!END OF TRANSACTION
C4341	C4710	D560	SRVN	TCON	outnup !remnants of skeleton after <br
T2050	C4341	VSNVSE	0	0	!<CR><NL>
C4342	C4710	D560	SRVN	TCON	outnup !variable skeleton
T3070	C4700	VSNVSE	0	0	!<c4070(t3300t3310)>END OF TRANSACTION
C4700	C4710	D40	UNVN	TCON	outnup !user skeleton data entry
T3030	C990	FNNVUE	0	0	!<c4710(t2050)>bye
T3040	C4800	FNNVUE	0	0	!<c4342>d (done with this skeleton)
T2520	C150	FNNVUE	0	0	!<c5071>done<CR>
T2610	C4380	FNNVUE	0	0	!<c5071>list<CR>
T2612	C4380	FNNVUE	0	0	!<c4360(t2050)>listl<CR>
T2640	C5020	FNNVUE	0	0	!<c4360(t2050)>old<CR>
T2550	C4710	FNNVUE	0	0	!remove clearfiles
T2555	C4710	FNNVUE	0	0	!remov clearfiles
T2560	C4710	FNNVUE	0	0	!remo clearfiles
T2630	C4711	FNNVUE	0	0	!<c5071>resave<CR>

T2631	C4711	FNNVUE	0	0	!<c4360(t2050)>resa<CR>
T3011	C4700	FNNIUE	0	0	!system<CR>
T3012	C4700	FNNIUE	0	0	!syst<CR>
T3000	C4370	FNNVUE	0	0	!test<CR>
T3001	C4370	FNNVUE	0	0	!t<CR>
T2890	C4370	FNNVUE	0	0	!tn<CR>
T3020	C4370	FNNVUE	0	0	!xmit<CR>
T3021	C4370	FNNVUE	0	0	!x<CR>
T3010	C4710	FNNVUE	0	0	!*<CR>
T2100	C4370	FNNVUE	0	0	!<CR>
T2960	C4370	VSNVUE	0	0	!backward
T2940	C4370	VSNVUE	0	0	!forward
T2980	C4370	VSNVUE	0	0	!get<SP>
T2540	C4800	VSNVUE	0	0	!dele(te)
T2541	C4700	VSNVUE	0	0	!feed(back)
T2547	C4700	VSNVUE	0	0	!help
T2611	C4380	VSNVUE	0	0	vumcl !list <filename>
T2613	C4380	VSNVUE	0	0	vumcl !listl <filename>
T2542	C4700	VSNVUE	0	0	!next
T2543	C4700	VSNVUE	0	0	!nofe(ed)
T2162	C4800	VSNVUE	0	0	vumc2 !old <filename>
T2544	C4700	VSNVUE	0	0	!pass
T2545	C4700	VSNVUE	0	0	!rewi(nd)
T2546	C4700	VSNVUE	0	0	!skip
T2548	C4700	VSNVUE	0	0	!voca(bulary)
T2632	C4713	VSNVUE	0	0	vumc2 !resave <filename>
T2633	C4713	VSNVUE	0	0	vumc2 !resa <filename>
T2580	C4710	VSNVUE	0	0	!rese(quence)
T2600	C4731	VSNVUE	0	0	!retr
T2161	C4710	VSNVUE	0	0	!ol <filename>
T2981	C4370	VSNVUE	0	0	!g<SP>
T2160	C4710	VSNVUE	0	0	!o <filename>
T2891	C4710	VSNVUE	0	0	!tn
T2910	C4710	VSNVUE	0	0	![
T2060	C4710	VSNVUE	0	0	!0<SP>
T2061	C4710	VSNVUE	0	0	!1<SP>
T2062	C4710	VSNVUE	0	0	!2<SP>
T2063	C4710	VSNVUE	0	0	!3<SP>
T2064	C4710	VSNVUE	0	0	!4<SP>
T2065	C4710	VSNVUE	0	0	!5<SP>
T2066	C4710	VSNVUE	0	0	!6<SP>
T2067	C4710	VSNVUE	0	0	!7<SP>
T2068	C4710	VSNVUE	0	0	!8<SP>
T2069	C4710	VSNVUE	U	0	!9<SP>
T2961	C4370	VSNVUE	0	0	!b
T2941	C4370	VSNVUE	0	0	!f
C4710	C4710	D570	SNVN	TCON	outnup !sys resp to skel data entry
T2050	C4700	FNNVSE	0	0	!<c4380(t2281)c4712c5050><CR><NL>
T3060	C4710	FNNVSE	0	0	!INVALID COMMAND AT THIS LEVEL

T3050	C4715	VSNVSE	0	0	! DATA TRUNCATED<SP>
C4711	C4710	D581	SNFN	TCON	outnup !sys resp to resa cmd
T2211	C4712	FNNVSE	0	0	!<all>FILE NAME?<SP>
C4712	C4370	D40	UNVN	TCON	outnup !user input filename
T2100	C4710	FNNVUE	0	0	!<all><CR>
T2100	C4713	VNFVUE	0	0	vumc1 !<all><filename>
C4713	C4710	D539	SNVN	TCON	outnup !sys resp to filename
T2270	C4714	FNNVSE	0	0	!<CR><NL>DATA SAVED-
T2196	C4710	VSNVSE	0	0	!FILE '<filename>' DOES NOT EXIST--USE
T2220	C4710	VSNVSE	0	0	!<CR><NL><50>FILE <filename> -- INCORRE
T2221	C4710	FNNVSE	0	0	!<CR><NL><NL>ERR-FILE NAME >8 CHARACTER
T2224	C4710	VSNVSE	0	0	!<CR><NL><NL>ERR-'<character>' IS AN IL
T2225	C4710	VSNVSE	0	0	!<CR><NL><NL>ERR-FIELD STARTING WITH '<
C4714	C4710	D560	SNVN	TCON	outnup !end of data saved msg
T2100	C4700	VNNVSE	0	0	!<filename><CR><NL>
C4715	C4710	D40	UNVN	TCON	outnup !  cr ^
T2100	C4710	VSNVUE	0	0	!anything
C4370	C4710	D580	SNVN	TCON	outnup ! ? ^E cr  nl ^R  cr  nl ^LEVE
T2240	C4710	VSNVSE	0	0	!TRANSACTION xxxxxx NOT IN CURR FILE<SP
T2657	C4371	FNNVSE	0	0	!<CR><NL>spaces<CR><NL>TN[000010spacesT
T2658	C4371	VSNVSE	0	0	!TRANSACTION SEQUENCE NUMBER <CR><NL>
T2659	C4371	VSNVSE	0	0	!ERROR MESSAGE<CR><NL>
T3090	C5060	FNNVSE	0	0	!TRANSACTION SENT. NEXT?<SP>
T3090	C5060	VSNVSE	0	0	!TRANSACTION SENT. NEXT?<SP>
T3060	C4710	FNNVSE	0	0	!INVALID COMMAND AT THIS LEVEL
C4371	C4710	D560	SRVN	TCON	outnup !  cr  nl ^
T3070	C4700	VSNVSE	0	0	!END OF TRANSACTION <CR><NL>
T3080	C4700	VSNVSE	0	0	!MORE - - - <CR><NL>
C4380	C4800	D506	SNVN	TCON	outnup !sys out of list
T2280	C4710	FNNVSE	0	0	!ready
T2281	C4710	VSNVSE	0	0	!<c4700(t2610t2612)>ready preceded by d
C4731	C4800	D598	SNVN	TCON	outnup !  cr  nl ^TYPE ^- ^
T2205	C4733	FNNVSE	0	0	!INVALID TRANSACTION TYPE ON RETRIEVE<C
T2240	C4710	VSNVSE	0	0	!TRANSACTION xxxxxx NOT IN CURR FILE<SP
T2250	C4710	FNNVSE	0	0	!<CR><NL>ILLEGAL PARAMETER<SP><SP><SP>
T2657	C4732	FNNVSE	0	0	!<CR><NL>spaces<CR><NL>TN[000010spacesT
T2658	C4732	VSNVSE	0	0	!TRANSACTION SEQUENCE NUMBER <CR><NL>
C4732	C4800	D560	SRVN	TCON	outnup ! <CR><NL>
T3070	C4700	VSNVSE	0	0	!END OF TRANSACTION <CR><NL>

C4733	C4731	D40	UNFN	TCON	!user enters tran type
T3300	C5070	FNNVUE	0	0	!hd
T3310	C5070	FNNVUE	0	0	!as
T3320	C5070	FNNVUE	0	0	!am
T3330	C5070	FNNVUE	0	0	!av
T3340	C5070	FNNVUE	0	0	!ae
T3350	C5070	FNNVUE	0	0	!at
T3380	C5070	FNNVUE	0	0	!ma
T3390	C5070	FNNVUE	0	0	!mc
T3400	C5070	FNNVUE	0	0	!pl
T3410	C5070	FNNVUE	0	0	!pd
T3420	C5070	FNNVUE	0	0	!ol
T3430	C5070	FNNVUE	0	0	!od
C4800	C5084	D540	SNVN	TCON	outnup !variable tran type
T2110	C4990	FNNVSE	0	0	!OLD OR NEW-
T2110	C4990	VSNVSE	0	0	!OLD OR NEW-
T2190	C4800	FNNVSE	0	0	!<c5030>READY
T2200	C5060	FNNVSE	0	0	!<c5050(t2190)><CR><NL><NL><CR><NL>TRAN
T2200	C5060	VSNVSE	0	0	!<CR><NL><NL><CR><NL>TRANSACTION TYPE ?
T2201	C5060	FNNVSE	0	0	!<c4010c4700(t3040)c5050(t2222)><CR><NL>
T2201	C5060	VSNVSE	0	0	!<CR><NL> TRANSACTION TYPE ?<SP><SP>
T2222	C4800	FNNVSE	0	0	!<c5030><CR><NL>ERR-FILE NAME >8 CHARAC
C4990	C5084	D40	UNVN	TCON	outnup !user response to old or new
T2130	C4020	FNNVUE	0	0	!new<CR>
T2140	C4020	FNNVUE	0	0	!n<CR>
T2640	C5020	FNNVUE	0	0	!old<CR>
T2645	C5020	FNNVUE	0	0	!o<CR>
T2162	C4800	VSFVUE	0	0	vumc2 !old <filename>
T2160	C4800	VSFVUE	0	0	vumc2 !o <filename>
T2180	C5012	FNNVUE	0	0	!same<CR>
T2181	C5012	FNNVUE	0	0	!s<CR>
T2100	C150	FNNVUE	0	0	!<CR>
C5012	C5084	D530	SNFN	TCON	outnup !sys resp to same
T2210	C4800	FNNVSE	0	0	!YOU PRESENTLY DO NOT HAVE A CURRENT FI
T2190	C4800	FNNVSE	0	0	!READY
C5020	C4800	D581	SNFN	TCON	outnup !sys resp to old<CR>
T3110	C5030	FNNVSE	0	0	!<c4030(t2640t2645)c4700(t2640)c4990(t2
C5030	C4800	D40	UNVN	TCON	outnup !user input old filename
T2100	C4800	FNNVUE	0	0	!<CR>
T2100	C4800	VNFVUE	0	0	vumc1 !<c5020><filename>
C5051	C4710	D560	SNVN	TCON	outnup !end of data saved msg
T2100	C4700	VNNVSE	0	0	!<filename><CR><NL>
C5060	C4710	D40	UNVN	TCON	outnup !user input tran type

T3300	C5070	FNNVUE	0	0	!<c5050(t2201)>hd
T3310	C5070	FNNVUE	0	0	!<c5050(t2201)>as
T3320	C5070	FNNVUE	0	0	!am
T3330	C5070	FNNVUE	0	0	!av
T3340	C5070	FNNVUE	0	0	!ae
T3350	C5070	FNNVUE	0	0	!at
T3380	C5070	FNNVUE	0	0	!ma
T3390	C5070	FNNVUE	0	0	!mc
T3400	C5070	FNNVUE	0	0	!pl
T3410	C5070	FNNVUE	0	0	!pd
T3420	C5070	FNNVUE	0	0	!ol
T3430	C5070	FNNVUE	0	0	!od
T3010	C5070	FNNVUE	0	0	!<c5050(t2201)>*
T3030	C990	FNNVUE	0	0	!bye
T2100	C5080	FNNVUE	0	0	!<CR>
T2540	C5062	VSNVUE	0	0	!dele(te) <transaction>
T2520	C150	FNNVUE	0	0	!done<CR>
T2610	C5061	FNNVUE	0	0	!list<CR>
T2612	C5061	FNNVUE	0	0	!listl<CR>
T2611	C5061	VS FVUE	0	0	vumcl !list <filename>
T2613	C5061	VS FVUE	0	0	vumcl !listl <filename>
T2640	C5020	FNNVUE	0	0	!old<CR>
T2162	C4800	VS FVUE	0	0	vumc2 !old <filename>
T2550	C5062	FNNVUE	0	0	!remove clearfiles
T2555	C5062	FNNVUE	0	0	!remov clearfiles
T2560	C5062	FNNVUE	0	0	!remo clearfiles
T2630	C5063	FNNVUE	0	0	!resave<CR>
T2631	C5063	FNNVUE	0	0	!resa<CR>
T2632	C5065	VS FVUE	0	0	vumc2 !resave <filename>
T2633	C5065	VS FVUE	0	0	vumc2 !resa <filename>
T2580	C5062	VSNVUE	0	0	!rese(quence)
T2600	C5070	VSNVUE	0	0	!retr
C5061	C5084	D506	SNVN	TCON	outnup !sys resp to list
T2280	C5062	FNNVSE	0	0	!ready
T2281	C5062	VSNVSE	0	0	!ready preceded by data/error
C5062	C5062	D1000	SNFN	TCON	outnup !sys resp after ready(list), re
T2201	C5060	FNNVSE	0	0	!<CR><NL> TRANSACTION TYPE ?<SP><SP>
T2202	C5060	FNNVSE	0	0	!<CR><NL>TRANSACTION TYPE ?<SP><SP>
C5063	C5062	D581	SNFN	TCON	outnup !sys resp to resave
T2211	C5064	FNNVSE	0	0	!FILE NAME?<SP>
C5064	C5062	D40	UNVN	TCON	outnup !user input filename
T2100	C5062	FNNVUE	0	0	!<CR>
T2100	C5065	VNFVUE	0	0	vumc1 !<all><filename>
C5065	C5062	D502	SNVN	TCON	outnup !sys resp to resave
T2202	C5060	VSNVSE	0	0	!<CR><NL>TRANSACTION TYPE ?<SP><SP> pre

T2221	C5062	FNNVSE	0	0	!<CR><NL><NL>ERR-FILE NAME >8 CHARACTER
C5070	C4800	D598	SNVN	TCON	outnup ! cr  nl ^CURR FILE ^
T2230	C5060	FNNVSE	0	0	!INVALID TRANSACTION TYPE -RETYPE-<SP><
T2205	C4733	FNNVSE	0	0	!INVALID TRANSACTION TYPE ON RETRIEVE<C
T2240	C5062	VSNVSE	0	0	!TRANSACTION xxxxxx NOT IN CURR FILE<SP
T2250	C5062	FNNVSE	0	0	!<CR><NL>ILLEGAL PARAMETER<SP><SP><SP>
T2657	C5071	FNNVSE	0	0	!<c5060(t3010)><CR><NL>spaces<CR><NL>TN
T2658	C5071	VSNVSE	0	0	!<c5060(t3300t3310)>TRANSACTION SEQUENC
C5071	C4341	D560	SRVN	TCON	outnup ! cr  nl ^
T3070	C4700	VSNVSE	0	0	!<all>END OF TRANSACTION <CR><NL>
C5080	C5084	D510	SNVN	TCON	outnup !tcon start5
T2002	C5090	FNNVSE	0	0	!TCON VERSION 6.3, JUNE 1976
T2020	C150	VSNVSE	0	0	!NO TEMPORARY FILE SPACE CODE
C5081	C5084	D501	SNFN	TCON	outnup !tcon first question
T2001	C5090	FNNVSE	0	0	!<all>FUNCTIONAL AREA?
C5084	C5084	D509	SNVN	TCON	outnup !back to system, top or type?
T150	C160	FNNVSE	0	0	!SYSTEM ?
T150	C160	VSNVSE	0	0	!SYSTEM ? preceded by message
T2000	C5083	FNNVSE	0	0	!TCON VERSION 6.3, JUNE 1976
T2201	C5060	FNNVSE	0	0	!<CR><NL> TRANSACTION TYPE ?<SP><SP>
C5083	C5084	D40	UNFN	TCON	outnup !user hits <nl> "
T2100	C5081	FNNVUE	0	0	!<all><CR>
C5090	C5084	D40	UNFN	TCON	outnup !user input functional area
T2030	C2800	FNNVUE	0	0	!<all>fs
T2040	C4800	FNNVUE	0	0	!<all>dem
T2100	C150	FNNVUE	0	0	!<all><CR>
C6000	C150	D6000	SNFA	SIOS	outnup ! begin sios
T6000	C6010	FNNVSE	0	0	! "SIOS ROUTINE ... "
C6010	C150	D40	SNVN	SIOS	outnup ! WWMCCS userid follows
T6010	C6020	VSNVSE	0	0	! "<userid>"
C6020	C150	D40	SRFN	SIOS	outnup ! prompt for function wanted
T6020	C6030	FNNVSE	0	0	! "WHAT FUNCTION DO YOU WANT ... "
C6030	C150	D40	UNVN	SIOS	outnup ! user enters function
T6030	C6100	VSNVUE	0	0	! m (move)
T6031	C6300	FNNVUE	0	0	! c (clear)
T6032	C6300	VSNVUE	0	0	! cl (clear)
T6033	C6400	VSNVUE	0	0	! s (search)
T6034	C6200	VSNVUE	0	0	! r (return)
T6035	C150	FNNVUE	0	0	! <carriage return>

C6100	C150	D40	SRFN	SIOS	outnup ! prompt for file to use
T6100	C6110	FNNVSE	0	0	! "SIOS MOVE ROUTINE ... "
C6110	C150	D40	UNVN	SIOS	outnup ! user enters file to use
T6110	C6120	VNFVUE	0	0	vumc1 ! <file name>
T6035	C150	FNNVUE	0	0	! <carriage return>
C6120	C150	D6120	SNFN	SIOS	outnup ! sios response to file name
T6120	C6130	FNNVSE	0	0	! "SYSTEM ID YOU WANT TO USE"
T6121	C6100	FNNVSE	0	0	! "YOUR FILE IS BUSY"
T6122	C6100	FNNVSE	0	0	! "UNABLE TO ACCESS YOUR FILE"
C6130	C150	D40	UNVN	SIOS	outnup ! user enters system id
T6110	C6140	VNNVUE	0	0	! <system id to use>
T6035	C150	FNNVUE	0	0	! <carriage return>
C6140	C150	D6140	SNFN	SIOS	outnup ! sios response to system id
T6140	C6150	FNNVSE	0	0	! "WUFI FILE YOU WANT TO USE"
T6141	C6120	FNNVSE	0	0	! "SYSTEM ID IS NOT AUTHORIZED"
C6150	C150	D40	UNVN	SIOS	outnup ! user enters wufi file
T6110	C6160	VNNVUE	0	0	! <wufi file name>
T6035	C150	FNNVUE	0	0	! <carriage return>
C6160	C150	D6160	SNFN	SIOS	outnup ! sios response to wufi file
T6160	C6170	FNNVSE	0	0	! "DATA WAS MOVED"
T6161	C6140	FNNVSE	0	0	! "SYSTEM ID IS INVALID ON WUFI"
C6170	C150	D40	SNVN	SIOS	outnup ! number of records moved
T6170	C6020	VSNVSE	0	0	! "nnnnnn RECORDS"
C6200	C150	D40	SRFN	SIOS	outnup ! prompt for file to use
T6200	C6210	FNNVSE	0	0	! "SIOS RETURN ROUTINE ... "
C6210	C150	D40	UNVN	SIOS	outnup ! user enters file to use
T6110	C6220	VNFVUE	0	0	vumc1 ! <file name>
T6035	C150	FNNVUE	0	0	! <carriage return>
C6220	C150	D6120	SNFN	SIOS	outnup ! sios response to file name
T6120	C6230	FNNVSE	0	0	! "SYSTEM ID YOU WANT TO USE"
T6121	C6200	FNNVSE	0	0	! "YOUR FILE IS BUSY"
T6122	C6200	FNNVSE	0	0	! "UNABLE TO ACCESS YOUR FILE"
C6230	C150	D40	UNVN	SIOS	outnup ! user enters system id
T6110	C6240	VNNVUE	0	0	! <system id to use>
T6035	C150	FNNVUE	0	0	! <carriage return>
C6240	C150	D6140	SNFN	SIOS	outnup ! sios response to system id
T6140	C6250	FNNVSE	0	0	! "WUFI FILE YOU WANT TO USE"



AD-A171 332

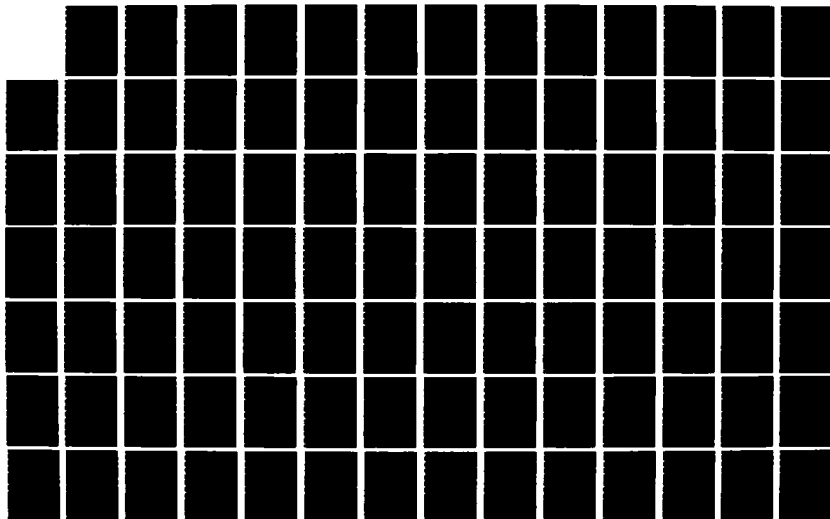
FORSOM SECURITY MONITOR (FSM) COMPUTER PROGRAM  
DEVELOPMENT SPECIFICATION (TYPE B5)(U) LOGICON INC SAN  
DIEGO CA FEB 81 N00039-78-C-0323

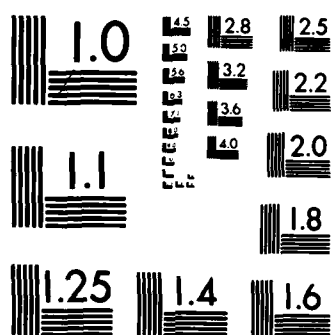
3/4

UNCLASSIFIED

F/G 9/2

ML





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

C6250	C150	D40	UNVN	SIOS	outnup ! user enters wufi file
T6110	C6260	VNNVUE	0	0	! <wufi file name>
T6035	C150	FNNVUE	0	0	! <carriage return>
C6260	C150	D6260	SNFN	SIOS	outnup ! sios response to wufi file
T6260	C6270	FNNVSE	0	0	! "RECORDS WERE RETURNED"
T6261	C6020	FNNVSE	0	0	! "NO DATA FOR THIS SYSTEM ID ... "
T2100	C6020	VSNVSE	0	0	!error message
C6270	C150	D40	SNVN	SIOS	outnup ! number of records returned
T6170	C6280	VSNVSE	0	0	! "nnnnnn RECORDS"
C6280	C150	D6140	SNFN	SIOS	outnup ! finish up normal return
T6280	C6020	FNNVSE	0	0	! "WUFI RECORDS WERE CLEARED"
C6300	C150	D40	SRFN	SIOS	outnup ! prompt for system id to use
T6300	C6310	FNNVSE	0	0	! "SIOS CLEAR ROUTINE ... "
C6310	C150	D40	UNVN	SIOS	outnup ! users enters system id
T6110	C6320	VNNVUE	0	0	! <system id>
T6035	C150	FNNVUE	0	0	! <carriage return>
C6320	C150	D6140	SNFN	SIOS	outnup ! sios response to system id
T6140	C6330	FNNVSE	0	0	! "WUFI FILE YOU WANT TO USE"
C6330	C150	D40	UNVN	SIOS	outnup ! user enters wufi file
T6110	C6340	VNNVUE	0	0	! <wufi file name>
T6035	C150	FNNVUE	0	0	! <carriage return>
C6340	C150	D6340	SNFN	SIOS	outnup ! sios response to wufi file
T6280	C6020	FNNVSE	0	0	! "WUFI RECORDS WERE CLEARED"
T6340	C6020	FNNVSE	0	0	! "NO DATA TO CLEAR"
C6400	C150	D40	SRFN	SIOS	outnup ! prompt for system id
T6400	C6410	FNNVSE	0	0	! "SIOS SEARCH ROUTINE ... "
C6410	C150	D40	UNVN	SIOS	outnup ! user enters system id
T6110	C6420	VNNVUE	0	0	! <system id>
T6035	C150	FNNVUE	0	0	! <carriage return>
C6420	C150	D6140	SNFN	SIOS	outnup ! sios response to system id
T6140	C6430	FNNVSE	0	0	! "WUFI FILE YOU WANT TO USE"
C6430	C150	D40	UNVN	SIOS	outnup ! user enters wufi file
T6110	C6440	VNNVUE	0	0	! <wufi file name>
T6035	C150	FNNVUE	0	0	! <carriage return>
C6440	C150	D6140	SNVN	SIOS	outnup ! sios response to wufi file
T6440	C6450	VSNVSE	0	0	! "nnnnnn BLOCKS OF DATA"
T6441	C6020	FNNVSE	0	0	! "NO RECORDS FOUND"

C6450	C150	D40	SNVN	SIOS	outnup ! number of records involved
T6170	C6020	VSNVSE	0	0	! "nnnnnn RECORDS"
C7000	C150	D7000	SRVA	LIST	outnup ! list output
T7000	C150	VSNVSE	0	0	! "ready"
T7001	C150	FNNVSE	0	0	!<CR><NL>illegal character
T7002	C150	FNNVSE	0	0	!<CR><NL><52>CURRENT FILE NOT DEFINED <
T150	C160	VSNVSE	0	0	!<CR><NL>SYSTEM ? (preceded by anything
C9000	C150	D20	SNFA	WWDMS	outnup !<pop>WWDMS entry point
T2110	C9010	FNNVSE	0	0	!<all>OLD OR NEW-
C9010	C150	D40	UNFN	WWDMS	outnup !user resp to old/new
T2100	C150	FNNVUE	0	0	!<c9000><CR>
T2130	C9020	FNNVUE	0	0	!<all>new<CR>
T2140	C9020	FNNVUE	0	0	!<all>n<CR>
C9020	C150	D9020	SNFN	WWDMS	outnup !<all>system is ready
T2190	C9030	FNNVSE	0	0	!<all>READY
C9030	C150	D9030	SNVN	WWDMS	outnup !WWDMS prompt
T9030	C9040	FNNVSE	0	0	!<c9020 c9210><CR><NL>*
T9030	C9040	VSNVSE	0	0	!<c9160(pop)>prompt preceded by error
C9040	C150	D40	UNFN	WWDMS	outnup !run commands
T9040	C9050	FNNVUE	0	0	!<c9030(t9030F)>>run mc76/exercise/equip
T9050	C9050	FNNVUE	0	0	!<c9030(t9030F)>>runs mc76/exercise/equi
T9060	C9050	FNNVUE	0	0	!<c9030(t9030F)>>runj mc76/exercise/equi
T9070	C9550	FNNVUE	0	0	!<c9030(t9030F)>>run mc76/exercise/rptmo
T9075	C9550	FNNVUE	0	0	!<c9030(t9030F)>>runs mc76/exercise/rptm
T9080	C9550	FNNVUE	0	0	!<c9030(t9030F)>>runj mc76/exercise/rptm
T9090	C9300	FNNVUE	0	0	!<c9030(t9030F)>>run mc76/exercise/accim
T9100	C9300	FNNVUE	0	0	!<c9210(t9030V)>>runs mc76/exercise/acci
T9110	C9300	FNNVUE	0	0	!<c9210(t9030V)>>runj mc76/exercise/acci
T9120	C9300	FNNVUE	0	0	!<c9170(t9030V)>>run mc76/exercise/info
T9130	C9300	FNNVUE	0	0	!<c9210(t9030V)>>runs mc76/exercise/info
T9140	C9300	FNNVUE	0	0	!<c9160(t9030F)>>runj mc76/exercise/info
T9150	C9300	FNNVUE	0	0	!<c9160(t9030F)>>run mc76/exercise/stren
T9160	C9300	FNNVUE	0	0	!runs mc76/exercise/strength
T9170	C9300	FNNVUE	0	0	!<c9030(t9030F)>>runj mc76/exercise/stre
T9180	C9300	FNNVUE	0	0	!run mc76/exercise/ready
T9190	C9300	FNNVUE	0	0	!runs mc76/exercise/ready
T9200	C9300	FNNVUE	0	0	!<c9030(t9030F)>>runj mc76/exercise/read
T9210	C9300	FNNVUE	0	0	!run mc76/exercise/pomcus
T9220	C9300	FNNVUE	0	0	!runs mc76/exercise/pomcus
T9230	C9300	FNNVUE	0	0	!<c9030(t9030F)>>runj mc76/exercise/pomc
T9240	C9550	FNNVUE	0	0	!run mc76/exercise/deployrp
T9250	C9550	FNNVUE	0	0	!runs mc76/exercise/deployrp
T9260	C9550	FNNVUE	0	0	!runj mc76/exercise/deployrp
T9270	C9300	FNNVUE	0	0	!run mc76/exercise/dplohpa

T9280	C9300	FNNVUE	0	0	!runs mc76/exercise/dployhpa
T9290	C9300	FNNVUE	0	0	!runj mc76/exercise/dployhpa
T9300	C9300	FNNVUE	0	0	!run mc76/exercise/cinc
T9310	C9300	FNNVUE	0	0	!runs mc76/exercise/cinc
T9320	C9300	FNNVUE	0	0	!<c9030(t9030F)>runj mc76/exercise/cinc
T9330	C9550	FNNVUE	U	0	!run mc76/exercise/exdeploy
T9340	C9550	FNNVUE	0	0	!runs mc76/exercise/exdeploy
T9350	C9550	FNNVUE	0	0	!runj mc76/exercise/exdeploy
T9360	C9300	FNNVUE	0	0	!run mc76/exercise/trans
T9370	C9300	FNNVUE	0	0	!runs mc76/exercise/trans
T9380	C9300	FNNVUE	0	0	!<c9030(t9030F)>runj mc76/exercise/tran
T9390	C9550	FNNVUE	0	0	!<c9030(t9030F)>run mc76/exercise/rptor
T9400	C9550	FNNVUE	0	0	!runs mc76/exercise/rptor
T9410	C9550	FNNVUE	0	0	!runj mc76/exercise/rptor
T9420	C9300	FNNVUE	0	0	!run mc76/exercise/mbcoistl
T9430	C9300	FNNVUE	0	0	!runs mc76/exercise/mbcoistl
T9440	C9300	FNNVUE	0	0	!<c9030(t9030F)>runj mc76/exercise/mbco
T3030	C990	FNNVUE	0	0	!<c9030>bye
T2520	C150	FNNVUE	0	0	!<c9030(t9030F)>done<CR>
T2100	C9030	FNNVUE	0	0	!<CR>
C9050	C9030	D9040	SNFN	WWDMS	outnup !compiler pass 1
T9500	C9051	FNNVSE	0	0	!<c9040(t9040 t9060)><CR><NL> .
C9051	C9030	D560	SNFN	WWDMS	outnup !compiler pass 2
T9501	C9052	FSNVSE	0	1	!<all>.<CR><NL>
C9052	C9030	D9045	SNFN	WWDMS	outnup !1st question for EQUIP
T9520	C9070	FNNVSE	0	0	!<all>ENTER REQUIRED PARAMETER FOR 'OMN
C9070	C9030	D40	UNVN	WWDMS	outnup !user enters cat file
T2100	C9080	VSNVUE	0	0	!<all>omni/<userid-prefix>/<filename>
C9080	C9030	D9060	SNVN	WWDMS	outnup !query for fldname
T9530	C9090	FNNVSE	0	0	!<all>ENTER FLDNAME = LITERAL
T9530	C9090	VSNVSE	0	0	!ENTER FLDNAME = LITERAL
T9030	C9040	FNNVSE	0	0	!<CR><NL>*
T9030	C9040	VSNVSE	0	0	!prompt preceded by error
C9090	C9030	D40	UNVN	WWDMS	outnup !user enters fieldname
T2100	C9100	VSNVUE	0	0	!<all>fieldname = "literal"
C9100	C9030	D9060	SNVN	WWDMS	outnup !query for sort-key1
T9540	C9110	FNNVSE	0	0	!<c9090>ENTER SORT-KEY1
T9540	C9110	VSNVSE	0	0	!ENTER SORT-KEY1
T9030	C9040	FNNVSE	0	0	!<CR><NL>*
T9030	C9040	VSNVSE	0	0	!prompt preceded by error
C9110	C9030	D40	UNVN	WWDMS	outnup !user enters sort-key1
T2100	C9120	VSNVUE	0	0	!<all>sort-key1

C9120	C9030	D9060	SNVN	WWDMS	outnup !query for sort-key2
T9550	C9130	FNNVSE	0	0	!<c9110>ENTER SORT-KEY2
T9550	C9130	VSNVSE	0	0	!ENTER SORT-KEY2
T9030	C9040	FNNVSE	0	0	!<CR><NL>*
T9030	C9040	VSNVSE	0	0	!prompt preceded by error
C9130	C9030	D40	UNVN	WWDMS	outnup !user enters last arg
T9325	C9140	VNNVUE	0	0	!<c9120(t9550F) c9390(t9600F t9600V)>la
C9140	C9030	D9060	SNVN	WWDMS	outnup !query for output file
T9560	C9150	FNNVSE	0	0	!<all>ENTER OUTPUT FILENAME
T9560	C9150	VSNVSE	0	0	!<all>ENTER OUTPUT FILENAME
T9030	C9040	FNNVSE	0	0	!<CR><NL>*
T9030	C9040	VSNVSE	0	0	!prompt preceded by error
C9150	C9030	D40	UNVN	WWDMS	outnup !user enters filename
T9325	C9160	VNFVUE	0	0	vumc1 !<c9140><filename>
C9160	C9160	D9050	SNVN	WWDMS	!compiler pass 3
T9500	C9170	FNNVSE	0	0	!<all><CR><NL> .
T9030	C9040	FNNVSE	0	0	!<all><CR><NL>*
T9030	C9040	VSNVSE	0	0	!prompt preceded by error
C9170	C9170	D9070	SNVN	WWDMS	!run-id
T9501	C9170	FSNVSE	0	1	!<c9160(t9500F)>.<CR><NL>
T9700	C9180	FNNVSE	0	0	!<c9170(t9501)>RUN-ID?
T9030	C9040	FNNVSE	0	0	!<CR><NL>*
T9030	C9040	VSNVSE	0	0	!<c9170(t2100)>prompt preceded by error
T2050	C9170	FNNVSE	0	0	!<c9160(t9500F)><CR><NL>
T9325	C9170	VNNVSE	0	0	!<c9170(t2050)>errors
C9180	C9030	D40	UNVN	WWDMS	outnup !user enters run-id
T9325	C9190	VNNVUE	0	0	!<all><run-id>
C9190	C9030	D9080	SNVN	WWDMS	outnup !ident?
T9710	C9200	FNNVSE	0	0	!<all>\$ IDENT?
T9750	C9170	FNNVSE	0	0	!<CR><NL>MUST BE 12 CHARACTERS OR LESS
T2100	C9030	VNNVSE	0	0	!(anything else)
C9200	C9030	D40	UNVN	WWDMS	outnup !user enters ident
T100	C9210	VSNVUE	0	0	!<all><ident>
C9210	C9211	D9120	SNVN	WWDMS	!end of status
T9030	C9040	FNNVSE	0	0	!<CR><NL>*
T9030	C9040	VSNVSE	0	0	!<c9200 c9210(t10290)><CR><NL>* precede
T10210	C9210	VSNVSE	0	0	!<c9200>(ab)normal termination
T10290	C9210	VSNVSE	0	0	!<c9200>NOT IN SYSTEM
T10300	C9030	VSNVSE	0	0	!<c9200>INVOKE JOUT MANUALLY
T9800	C9230	VSNVSE	0	0	!<c9210(t10210)>JOUT INVOKED FOR SNUMB.
T2100	C9210	FNNVSE	0	0	! <carriage return> (only)

T9325	C9210	VNNVSE	0	0	! anything else
C9211	C9211	D9061	SNVN	WWDMS	outnup !<break> during output
T9030	C9040	FNNVSE	0	0	!<CR><NL>*
T9030	C9040	VSNVSE	0	0	!<CR><NL>* preceded by error
T9755	C9212	FNNVSE	0	0	!LOOK, MORE, DELE, ABRT, OR QUIT?
C9212	C9211	D40	UNVN	WWDMS	outnup !user response
T9760	C9210	FNNVUE	0	0	!look
T9761	C9213	FNNVUE	0	0	!more
T9762	C9215	VSNVUE	0	0	!dele(te)
T9763	C9218	FNNVUE	0	0	!abrt
T9764	C9210	FNNVUE	0	0	!quit
T2100	C9210	FNNVUE	0	0	!<CR>
C9213	C9211	D9097	SNFN	WWDMS	outnup !asks for snumbs
T9756	C9214	FNNVSE	0	0	!SNUMB(S)-
C9214	C9211	D40	UNVN	WWDMS	outnup !user inputs snumbs
T2100	C9210	VSNVUE	0	0	!numbers or <CR>
C9215	C9211	D581	SNFN	WWDMS	outnup !asks for snumbs
T9757	C9216	FNNVSE	0	0	!SNUMB?
C9216	C9211	D40	UNVN	WWDMS	outnup !user inputs snumb
T2100	C9211	FNNVUE	0	0	!<CR>
T2100	C9217	VSNVUE	0	0	!number
C9217	C9211	D9063	SNFN	WWDMS	outnup !bad snumb
T9758	C9215	FNNVSE	0	0	!SNUMB NOT IN TABLE
T2100	C9211	VNNVSE	0	0	!errors?
C9218	C9211	D9064	SNFN	WWDMS	outnup !asks for snumb
T9759	C9219	FNNVSE	0	0	!snumb?
C9219	C9211	D40	UNVN	WWDMS	outnup !user inputs snumb
T2100	C9220	VSNVUE	0	0	!number or <CR>
C9220	C9211	D9065	SNFN	WWDMS	outnup !bad snumb
T9754	C9211	FNNVSE	0	0	!SNUMB NOT IN SYSTEM
T2100	C9211	VNNVSE	0	0	!errors?
C9230	C9231	D9100	SNVN	WWDMS	outnup !asks for print func
T10270	C9030	VSNVSE	0	0	!output not found
T10280	C9030	VSNVSE	0	0	!output busy
T9801	C9240	FNNVSE	0	0	!<c9210(t9800)>function?
T9030	C9040	FNNVSE	0	0	!<CR><NL>*
C9231	C9231	D9096	SNVN	WWDMS	outnup !user must release, etc.
T9030	C9040	FNNVSE	0	0	!<CR><NL>*

T9030	C9040	VSNVSE	0	0	!<CR><NL>* preceded by data/error
T9780	C9270	FNNVSE	0	0	!please direct, release, or hold before
C9240	C9231	D40	UNFN	WWDMS	outnup !user inputs print
T9802	C9250	FNNVUE	0	0	!print \$\$
T9803	C9250	FNNVUE	0	0	!prin \$\$
T10230	C9250	FNNVUE	0	0	!<all>print 74
T10240	C9250	FNNVUE	0	0	!prin 74
C9250	C9231	D9095	SRVN	WWDMS	outnup !end of print
T10170	C9260	VSNVSE	0	0	!end of \$\$
T10250	C9260	VSNVSE	0	0	!end of 74
T10260	C9230	FSNVSE	2	13	!74 not found
C9260	C9231	D9064	SNFN	WWDMS	outnup !second function
T9801	C9270	FNNVSE	0	0	!function?
C9270	C9231	D40	UNVN	WWDMS	outnup !second func choices
T10180	C9030	FNNVUE	0	0	!hold
T10190	C9030	VSNVUE	0	0	!rele
T10200	C9030	FNNVUE	0	0	!direct ac
T10205	C9030	FNNVUE	0	0	!direct onl
C9300	C9030	D9040	SNFN	WWDMS	outnup !compiler pass 1
T9500	C9301	FNNVSE	0	0	!<c9040(t9110 t9140 t9440 t9200 t9230 t
C9301	C9030	D560	SNFN	WWDMS	outnup !compiler pass 2
T9501	C9302	FSNVSE	0	1	!<all>.<CR><NL>
C9302	C9030	D9045	SNFN	WWDMS	outnup !1st question for ACCIMOB, INFO
T9520	C9320	FNNVSE	0	0	!<all>ENTER REQUIRED PARAMETER FOR 'OMN
C9320	C9030	D40	UNVN	WWDMS	outnup !user enters cat file
T9325	C9330	VNNVUE	0	0	!<all>omni/<userid-prefix>/<filename>
C9330	C9030	D9060	SNVN	WWDMS	outnup !query for fldname
T9530	C9340	FNNVSE	0	0	!<all>ENTER FLDNAME = LITERAL
T9530	C9340	VSNVSE	0	0	!ENTER FLDNAME = LITERAL
T9531	C9340	FNNVSE	0	0	!<all>ENTER FIELDNAME = LITERAL
T9531	C9340	VSNVSE	0	0	!<all>ENTER FIELDNAME = LITERAL
T9030	C9040	FNNVSE	0	0	!<CR><NL>*
T9030	C9040	VSNVSE	0	0	!<CR><NL>* preceded by error
C9340	C9030	D40	UNVN	WWDMS	outnup !user enters fieldname
T9325	C9350	VNNVUE	0	0	!<all>fieldname = "literal"
C9350	C9030	D9060	SNVN	WWDMS	outnup !query for sort-field1
T9580	C9360	FNNVSE	0	0	!<all>ENTER SORT-FLD1
T9580	C9360	VSNVSE	0	0	!<c9340>ENTER SORT-FLD1
T9030	C9040	FNNVSE	0	0	!<CR><NL>*



T9030	C9040	VSNVSE	0	0	!<CR><NL>* preceded by error
C9360	C9030	D40	UNVN	WWDMS	outnup !user enters sort-field1
T9325	C9370	VNNVUE	0	0	!<all>sort-field1
C9370	C9030	D9060	SNVN	WWDMS	outnup !query for sort-field2
T9590	C9380	FNNVSE	0	0	!<all>ENTER SORT-FLD2
T9590	C9380	VSNVSE	0	0	!<all>ENTER SORT-FLD2
T9030	C9040	FNNVSE	0	0	!<CR><NL>*
T9030	C9040	VSNVSE	0	0	!<CR><NL>* preceded by error
C9380	C9030	D40	UNVN	WWDMS	outnup !user enters sort-field2
T9325	C9390	VNNVUE	0	0	!<all>sort-field2
C9390	C9030	D9060	SNVN	WWDMS	outnup !sort-field3 query
T9600	C9130	FNNVSE	0	0	!<all>ENTER SORT-FLD3
T9600	C9130	VSNVSE	0	0	!<all>ENTER SORT-FLD3
T9030	C9040	FNNVSE	0	0	!<CR><NL>*
T9030	C9040	VSNVSE	0	0	!<CR><NL>* preceded by error
C9550	C9030	D9040	SNFN	WWDMS	outnup !compiler pass 1
T9500	C9551	FNNVSE	0	0	!<CR><NL> .
C9551	C9030	D560	SNFN	WWDMS	outnup !compiler pass 2
T9501	C9552	FSNVSE	0	1	!<CR><NL>
C9552	C9030	D9045	SNFN	WWDMS	outnup !1st question for RPTMOB,DEPLOY
T9570	C9570	FNNVSE	0	0	!ENTER PARAM FOR 'SBRPT'
C9570	C9030	D40	UNVN	WWDMS	outnup !user enters sbrpt
T2100	C9580	VSNVUE	0	0	!<all>"w..."
C9580	C9030	D9060	SNVN	WWDMS	outnup !query for cat file
T9521	C9590	FNNVSE	0	0	!<all>ENTER PARAM FOR OMNI CATFILE
T9521	C9590	VSNVSE	0	0	!ENTER PARAM FOR OMNI CATFILE
T9030	C9040	FNNVSE	0	0	!<CR><NL>*
T9030	C9040	VSNVSE	0	0	!<CR><NL>* preceded by error
C9590	C9030	D40	UNVN	WWDMS	outnup !user enters catfile name
T2100	C9350	VSNVUE	0	0	!<all>omni/<userid-prefix>/<filename>
C11000	C999	D11000	SNVN	TLCF	outnup ! announce teleconferencing
T11000	C11010	VSNVSE	0	0	! "TELECONFERENCING AT ... "
T11001	C280	FNNVSE	0	0	! "ILLEGAL CLASSIFICATION CODE"
C11010	C11010	D11010	SNFN	TLCF	outnup ! prompt user for conf action
T11010	C11020	FSNVSE	122	31	! "USER ASSISTANCE IS AVAILABLE ... "
C11020	C11010	D40	UNFN	TLCF	outnup ! user enters conf action
T11020	C11030	FNNVUE	0	U	! j(oin)

T11021	C11030	FNNVUE	0	0	! jo(in)
T11022	C11030	FNNVUE	0	0	! joi(n)
T11023	C11030	FNNVUE	0	0	! join
C11030	C11010	D11030	SNFN	TLCF	outnup ! prompt user for conf name
T11030	C11040	FNNVSE	0	0	! "NAME OF CONFERENCE YOU WISH TO JOIN?"
T11031	C999	VSVFSE	0	0	exit ! "LINE TERMINATED" (bad conf n
C11040	C11010	D40	UNFN	TLCF	outnup ! user enters conf name
T11040	C11050	VNNVUE	0	0	! <conference name>
C11050	C11030	D11050	SNFN	TLCF	outnup ! prompt user for name
T11050	C11060	FNNVSE	0	0	! "PARTICIPANT'S NAME? "
T11051	C11030	FNNVSE	0	0	! "SORRY, THAT CONFERENCE IS NOT IN SES
C11060	C11010	D40	UNVN	TLCF	outnup ! user enters name
T11040	C11070	VNNVUE	0	0	! <name>
C11070	C11050	D11070	SNFN	TLCF	outnup ! prompt user for password
T11070	C11080	FNNVSE	0	0	! "PARTICIPANT'S PASSWORD ... "
T11071	C11050	FNNVSE	0	0	! "SORRY, NAME GIVEN NOT RECOGNIZED"
T11072	C999	FNNVSE	0	0	! "ACCESS DENIED"
C11080	C11010	D40	UNVN	TLCF	outnup ! user enters password
T11040	C11090	VNNVUE	0	0	! <participant password>
C11090	C11070	D11090	SNFN	TLCF	outnup ! prompt user for host name
T11090	C11100	FNNVSE	0	0	! "THE FOLLOWING QUESTIONS CONCERN ... "
T11091	C11140	FNNVSE	0	0	! "ACCESS GRANTED"
T11072	C999	FNNVSE	0	0	! "ACCESS DENIED"
T11070	C11080	FNNVSE	0	0	! "PARTICIPANTS PASSWORD ... "
C11100	C11010	D40	UNVN	TLCF	outnup ! user enters host name
T11040	C11110	VNNVUE	0	0	! <host name>
C11110	C11090	D11110	SNFN	TLCF	outnup ! prompt user for ident
T11110	C11120	FNNVSE	0	0	! "ENTER YOUR ACCOUNT NUMBER, ... "
T11111	C11120	FNNVSE	0	0	! "ENTER YOUR ACCOUNT NUMBER, ... "
T12271	C11090	FNNVSE	0	0	!INVALID HOST NAME<SP><SP><SP>
C11120	C11010	D40	UNVN	TLCF	outnup ! user enters ident
T11120	C11130	VSNVUE	0	0	! <ident>
C11130	C11010	D11130	SNFN	TLCF	outnup ! outcome of join
T11091	C11140	FNNVSE	0	0	! "ACCESS GRANTED"
C11140	C11010	D11140	SNVN	TLCF	outnup ! conf security classification
T11140	C11150	VSNVSE	0	0	! "CONFERENCE SECURITY CLASSIFICATION .
C11150	C11010	D11140	SNVN	TLCF	outnup ! message announcement status

T11150	C11160	FNNVSE	0	0	! "THERE ARE NO ANNOUNCEMENTS AT THIS TI
T11151	C11160	VSNVSE	0	0	! "ANNOUNCEMENT MESSAGE NUMBERS ARE ...
C11160	C11010	D11140	SNVN	TLCF	outnup ! current message status
T11160	C11170	VSNVSE	0	0	! "LATEST MESSAGE NUMBER IS n"
C11170	C11010	D11140	SNVN	TLCF	outnup ! previous message status
T11170	C11180	VSNVSE	0	0	! "LAST MESSAGE SEEN WAS NUMBER n"
C11180	C12000	D11180	SNVN	TLCF	outnup ! listen mode entered initially
T11180	C11510	VSNVSE	0	0	! "LISTEN MODE ENTERED IN ... "
C11500	C12000	D11180	SNVN	TLCF	outnup ! listen mode entered subsequen
T11500	C11510	VSNVSE	0	0	! "LISTEN MODE ENTERED IN ... "
C11510	C12000	D11510	BRVN	TLCF	! listen mode polling
T11510	C11510	VSNVSE	0	0	!<NP>
T11190	C11510	VSNVSE	0	0	!<dtg> <name> HAS JOINED THE CONFERENCE
T11191	C11510	VSNVSE	0	0	!<dtg> <name> HAS LEFT THE CONFERENCE<S
T11040	C11510	VNNVUE	0	0	! (any user input is accepted and ignor
C12000	C12000	D12000	SNVN	TLCF	outnup ! command mode entered
T12000	C12010	FNNVSE	0	0	! "COMMAND ?"
T12001	C12010	FNNVSE	0	0	! "COMMAND ?" (preceded by form feed)
T12000	C12010	VSNVSE	0	0	! "COMMAND ?" (preceded by message)
C12010	C12000	D40	UNVN	TLCF	outnup ! user enters command
T12022	C12500	VSNVUE	0	0	! desc(ribe)
T12012	C12100	VSNVUE	0	0	! defi(nition)
T12010	C12040	VSNVUE	0	0	! bull(etin)
T12011	C12060	VSNVUE	0	0	! comm(unication)
T12013	C12200	VSNVUE	0	0	! floo(r)
T12014	C12240	VSNVUE	0	0	! give(up)
T12015	C11500	VSNVUE	0	0	! list(en)
T12016	C12250	VSNVUE	0	0	! prin(t)
T12017	C12350	VSNVUE	0	0	! quit
T12018	C12360	VSNVUE	0	0	! revi(ew)
T12019	C12450	VSNVUE	0	0	! stat(us)
T12020	C13000	VSNVUE	0	0	! talk
T12021	C12470	VSNVUE	0	0	! ?
C12040	C12000	D12040	SNFN	TLCF	outnup ! bulletin verb
T12040	C12000	FSNVSE	0	21	! "NO MESSAGES WAITING"
T12041	C12050	FNNVSE	0	0	! "MESSAGE NUMBER"
C12050	C12000	D12050	SRVN	TLCF	outnup ! bulletin message
T11600	C12050	VSNVSE	0	0	!<CR><NL><NP>
T12050	C12010	VSNVSE	0	0	! "COMMAND ?"
C12060	C12000	D12060	SNFN	TLCF	outnup ! communication

T12060	C12070	FSNVSE	0	92	! "ENTER ONE LINE PER PROMPT (-) ... "
T12061	C12000	FSNVSE	0	34	! "INVALID COMMUNICATION ARGUMENT"
C12070	C12000	D12070	SNVN	TLCF	outnup ! comm input prompt
T12070	C12080	FSNVSE	0	3	! "-"
T12071	C12000	FSNVSE	0	22	! "COMMUNICATION SENT"
T12072	C12000	FSNVSE	0	34	! "COMMUNICATION REQUEST IGNORED"
T12000	C12010	VSNVSE	0	0	! "COMMAND ?" (preceded by msg)
C12080	C12000	D40	UNFN	TLCF	outnup ! user comm input
T11040	C12070	VNNVUE	0	0	! (any user input)
C12100	C12000	D12100	SRVN	TLCF	outnup ! definition output
T12100	C12010	FNNVSE	0	0	! "FLOOR RECORDING IS OFF"
T12101	C12010	FNNVSE	0	0	! "FLOOR RECORDING IS ON"
T12102	C12010	FNNVSE	0	0	! "CONFERENCE IS OPEN TO JOINING ... "
T12103	C12010	FNNVSE	0	0	! "CONFERENCE IS CLOSED TO JOINING ... "
T12104	C12010	FNNVSE	0	0	! "OPTION RESTRICTED TO INITIATING ... "
T12105	C12010	FNNVSE	0	0	! "OPTION RESTRICTED TO CURRENT ... "
T12106	C12010	FNNVSE	0	0	! "NO ALTERNATES HAVE BEEN ASSIGNED"
T12107	C12110	VSNVSE	0	0	! " ... CHANGE?"
T12108	C12100	VSNVSE	0	0	! " ... FILE INSERT LIMIT IS— ... "
T12000	C12010	VSNVSE	0	0	! "COMMAND ?" (preceded by anything)
C12110	C12000	D40	UNVN	TLCF	outnup ! user responds to definition p
T11040	C12120	VNNVUE	0	0	! user response
C12120	C12000	D12120	SNVN	TLCF	outnup ! system response to user input
T12000	C12010	FNNVSE	0	0	! "COMMAND ?" (only)
T12120	C12110	FSNVSE	0	12	! "HOST?"
T12121	C12110	FSNVSE	0	11	! "\$IDENT?"
T12122	C12110	FSNVSE	0	54	! "NEW IDENT CARD NOT VALID AT THIS ... "
T12000	C12010	VSNVSE	0	0	! "COMMAND ?" (preceded by anything)
C12200	C12000	D12200	SNVN	TLCF	outnup ! floor response
T12200	C12210	FSNVSE	0	53	! "ENTER ONE LINE OF TEXT ... "
T12201	C12000	FSNVSE	0	54	! "CHAIRMAN NOT PRESENT ... "
T12202	C12000	FSNVSE	0	30	! "FLOOR INVALID COMMAND"
T12203	C12010	VSNVSE	0	0	! "INVALID FLOOR COMMAND ... "
T12000	C12010	VSNVSE	0	0	! "COMMAND ?" (preceded by anything)
C12210	C12000	D12210	SNFN	TLCF	outnup ! floor prompt for input
T12210	C12220	FSNVSE	0	62	! "(ENTER BREAK STATUS OR NULL ...)"
C12220	C12000	D40	UNVN	TLCF	outnup ! floor request input
T11040	C12230	VNNVUE	0	0	! <line of input to accompany floor req
C12230	C12000	D12230	SNFN	TLCF	outnup ! floor request acknowledged
T12230	C12000	FSNVSE	0	64	! "CHAIRMAN'S APPROVAL ... "

C12240	C12000	D12240	SNFN	TLCF	outnup ! giveup response
T12240	C12000	FSNVSE	0	30	! "COMMAND VALID FOR FLOOR ONLY"
C12250	C12000	D12250	SNVN	TLCF	outnup ! print response
T12250	C12000	FSNVSE	0	38	! "PRINT REQUEST INCOMPLETE OR INVALID"
T12251	C12000	FSNVSE	0	26	! "MESSAGE NUMBER TOO HIGH"
T12252	C12260	FSNVSE	0	0	! " ... CHANGE?"
T12000	C12010	VSINVSE	0	0	! "COMMAND ?" (preceded by anything)
C12260	C12000	D40	UNVN	TLCF	outnup ! user enters info to prompt
T11040	C12270	VNNVUE	0	0	! (any user input)
C12270	C12000	D12270	SNVN	TLCF	outnup ! print request prompts
T12270	C12260	FSNVSE	0	46	! "ENTER HOST NAME FOR DELIVERY ... "
T12271	C12270	FSNVSE	0	22	! "INVALID HOST NAME"
T12272	C12260	VSINVSE	0	0	! " ... CHANGE? "
T12273	C12260	FSNVSE	0	52	! "ENTER YOUR ACCOUNT NUMBER, NAME ... "
T12274	C12260	FSNVSE	0	80	! "IDENT INFORMATION NOT VALID ... "
T12000	C12010	VSINVSE	0	0	! "COMMAND ?" (preceded by anything)
C12350	C12000	D12350	SNFN	TLCF	outnup ! quit
T12350	C999	FSNVSE	0	18	! "QUIT SUCCESSFUL"
C12360	C12000	D12360	SNFN	TLCF	outnup ! review
T12360	C12000	FSNVSE	0	24	! "INVALID REVIEW REQUEST"
T11600	C12050	FNNVSE	0	0	!<CR><NL><NP>
C12450	C12000	D12450	SRVN	TLCF	outnup ! status
T12050	C12010	VSINVSE	0	0	! "COMMAND ?"
C12470	C12000	D12450	SRFN	TLCF	outnup ! ? output
T12470	C12010	FSNVSE	809	9	! "THIS TERMINAL IS CURRENTLY ... "
C12500	C12000	D12500	SNFN	TLCF	outnup ! determine describe output
T12500	C12600	FSNVSE	5	14	! ">>>ADD<<<"
T12501	C12610	FSNVSE	5	14	! ">>>ADJOURN<<<"
T12502	C12620	FSNVSE	5	14	! ">>>ANNOUNCE<<<"
T12503	C12630	FSNVSE	5	14	! ">>>BULLETIN<<<"
T12504	C12640	FSNVSE	5	14	! ">>>CHAIR TO<<<"
T12505	C12650	FSNVSE	5	14	! ">>>COMMUNICATION<<<"
T12506	C12660	FSNVSE	5	14	! ">>>DEFINITION<<<"
T12507	C12670	FSNVSE	5	14	! ">>>DELETE<<<"
T12508	C12680	FSNVSE	5	14	! ">>>DESCRIBE<<<"
T12509	C12690	FSNVSE	5	14	! ">>>ERASE<<<"
T12510	C12700	FSNVSE	5	14	! ">>>FLOOR<<<"
T12511	C12710	FSNVSE	5	14	! ">>>FORMAT<<<"
T12512	C12720	FSNVSE	5	14	! ">>>GIVEUP<<<"
T12513	C12730	FSNVSE	5	14	! ">>>INSERT<<<"
T12514	C12740	FSNVSE	5	14	! ">>>KEYWORD<<<"
T12515	C12750	FSNVSE	5	14	! ">>>LISTEN<<<"

T12516	C12760	FSNVSE	5	14	! ">>>PDAC<<<"
T12517	C12770	FSNVSE	5	14	! ">>>PRINT<<<"
T12518	C12780	FSNVSE	5	14	! ">>>QUIT<<<"
T12519	C12790	FSNVSE	5	14	! ">>>REVIEW<<<"
T12520	C12800	FSNVSE	5	14	! ">>>STATUS<<<"
T12521	C12810	FSNVSE	5	14	! ">>>TALK<<<"
T12522	C12820	FSNVSE	5	14	! ">>>TERMINATE<<<"
T12523	C12830	FSNVSE	5	14	! ">>>?<<<"
T12524	C12470	FSNVSE	5	14	! "INVALID DESCRIBE ARGUMENT PASSED"
C12600	C12000	D12450	SRFN	TLCF	outnup ! add text
T12600	C12010	FSNVSE	471	9	! "THIS VERB'S USAGE IS RESTRICTED ...
C12610	C12000	D12450	SRFN	TLCF	outnup ! adjourn text
T12610	C12010	FSNVSE	454	9	! "THIS VERB'S USAGE IS RESTRICTED ...
C12620	C12000	D12450	SRFN	TLCF	outnup ! announce text
T12620	C12010	FSNVSE	589	9	! "THIS VERB'S USAGE IS RESTRICTED ...
C12630	C12000	D12450	SRFN	TLCF	outnup ! bulletin text
T12630	C12010	FSNVSE	240	9	! "THIS VERB ALLOWS AN INDIVIDUAL ... "
C12640	C12000	D12450	SRFN	TLCF	outnup ! chair to text
T12640	C12010	FSNVSE	822	9	! "THIS VERB'S USAGE IS RESTRICTED ...
C12650	C12000	D12450	SRFN	TLCF	outnup ! communication text
T12650	C12010	FSNVSE	918	9	! "THIS VERB WILL ALLOW INFORMAL ... "
C12660	C12000	D12450	SRFN	TLCF	outnup ! definition text
T12660	C12010	FSNVSE	2644	9	! "THIS VERB DISPLAYS THE STATE ... "
C12670	C12000	D12450	SRFN	TLCF	outnup ! delete text
T12670	C12010	FSNVSE	360	9	! "THIS COMMAND ALLOWS THE ... "
C12680	C12000	D12450	SRFN	TLCF	outnup ! describe text
T12680	C12010	FSNVSE	572	9	! "THIS VERB IS USED TO OBTAIN ... "
C12690	C12000	D12450	SRFN	TLCF	outnup ! erase text
T12690	C12010	FSNVSE	289	9	! "THIS COMMAND IS USED BY THE ... "
C12700	C12000	D12450	SRFN	TLCF	outnup ! floor text
T12700	C12010	FSNVSE	1311	9	! "THE PURPOSE OF THE FLOOR ... "
C12710	C12000	D12450	SRFN	TLCF	outnup ! format text
T12710	C12010	FSNVSE	685	9	! "THIS COMMAND IS USED BY THE ... "
C12720	C12000	D12450	SRFN	TLCF	outnup ! giveup text
T12720	C12010	FSNVSE	181	9	! "THIS VERB ALLOWS A PARTICIPANT ... "
C12730	C12000	D12450	SRFN	TLCF	outnup ! insert text

T12730	C12010	FSNVSE	618	9	! "THIS IS A CHAIRMAN ONLY VERB ... "
C12740	C12000	D12450	SRFN	TLCF	outnup ! keyword text
T12740	C12010	FSNVSE	528	9	! "MESSAGES MAY BE ASSIGNED A ... "
C12750	C12000	D12450	SRFN	TLCF	outnup ! listen text
T12750	C12010	FSNVSE	328	9	! "TO RECEIVE CONFERENCE MESSAGES ... "
C12760	C12000	D12450	SRFN	TLCF	outnup ! pdac text
T12760	C12010	FSNVSE	561	9	! "THIS VERB OPENS A CONNECTION ... "
C12770	C12000	D12450	SRFN	TLCF	outnup ! print text
T12770	C12010	FSNVSE	1868	9	! "THIS VERB ALLOWS AN INDIVIDUAL ... "
C12780	C12000	D12450	SRFN	TLCF	outnup ! quit text
T12780	C12010	FSNVSE	115	9	! "THIS VERB DISCONNECTS A ... "
C12790	C12000	D12450	SRFN	TLCF	outnup ! review text
T12790	C12010	FSNVSE	1265	9	! "THIS VERB ALLOWS A SELECTIVE ... "
C12800	C12000	D12450	SRFN	TLCF	outnup ! status text
T12800	C12010	FSNVSE	294	9	! "THIS VERB PRODUCES A REPORT ... "
C12810	C12000	D12450	SRFN	TLCF	outnup ! talk text
T12810	C12010	FSNVSE	233	9	! "THIS VERB TRANSFERS A ... "
C12820	C12000	D12450	SRFN	TLCF	outnup ! terminate text
T12820	C12010	FSNVSE	422	9	! "THIS VERB'S USAGE IS RESTRICTED ... "
C12830	C12000	D12450	SRFN	TLCF	outnup ! ? text
T12830	C12010	FSNVSE	173	9	! "THE QUESTION MARK IS USED ... "
C13000	C13500	D13000	SNVN	TLCF	outnup ! talk mode entered
T13000	C13010	VSNVSE	0	0	! "TALK MODE ENTERED IN ... "
C13010	C13500	D13010	SNFN	TLCF	outnup ! talk mode prompt
T13010	C13050	FSNVSE	0	3	! ">" (talk mode prompt)
T13011	C13050	FSNVSE	0	3	! "-" (alternate talk mode prompt)
T13012	C13020	FNNVSE	0	0	!<CR><NL>T0?
T13012	C13020	VSNVSE	0	0	!<CR><NL>T0?
T13013	C13020	FSNVSE	U	7	! "FROM?"
T13014	C13020	FSNVSE	0	10	! "SUBJECT?"
T13015	C13020	FSNVSE	0	8	! "CLASS?"
T13016	C13020	FSNVSE	0	7	! "INFO?"
T13017	C13020	FSNVSE	0	6	! "REF?"
C13020	C13500	D40	UNVN	TLCF	outnup ! user enters format info
T11040	C13030	VNNVUE	0	0	! (anything is valid)
C13030	C13500	D13010	SNFN	TLCF	outnup ! complete message formatting

T13010	C13050	FSNVSE	0	3	! ">" (talk mode prompt)
T13011	C13020	FSNVSE	0	3	! "-" (msg format prompt)
T13012	C13020	FSNVSE	0	5	! "TO?"
T13013	C13020	FSNVSE	0	7	! "FROM?"
T13014	C13020	FSNVSE	0	10	! "SUBJECT?"
T13015	C13020	FSNVSE	0	8	! "CLASS?"
T13016	C13020	FSNVSE	0	7	! "INFO?"
T13017	C13020	FSNVSE	0	6	! "REF?"
C13050	C13500	D40	UNVN	TLCF	outnup ! user enters talk mode command
T13050	C13100	VSNVUE	0	0	! \$desc(ribe)
T13051	C12000	VSNVUE	0	0	! \$comm(and)
T13052	C13300	VSNVUE	0	0	! \$dele(te)
T13053	C13320	VSNVUE	0	0	! \$end
T12080	C13320	FNNVUE	0	0	! <carriage return> (same as \$end)
T13054	C13340	VSNVUE	0	0	! \$list
T13055	C13360	VSNVUE	0	0	! \$sele(ct)
T13056	C13380	VSNVUE	0	0	! \$subj(ect)
T13057	C13410	VSNVUE	0	0	! \$talk
T13058	C13210	FNNVUE	0	0	! ?
T11040	C13010	VNNVUE	0	0	! (anything else is part of a message)
C13100	C13500	D13100	SNFN	TLCF	outnup ! determine describe output
T13100	C13110	FSNVSE	5	14	! ">>>\$DESCRIBE<<<"
T13101	C13120	FSNVSE	5	14	! ">>>\$COMMAND<<<"
T13102	C13130	FSNVSE	5	14	! ">>>\$DELETE<<<"
T13103	C13140	FSNVSE	5	14	! ">>>\$END<<<"
T13104	C13150	FSNVSE	5	14	! ">>>\$LIST<<<"
T13105	C13160	FSNVSE	5	14	! ">>>\$SELECT<<<"
T13106	C13170	FSNVSE	5	14	! ">>>\$SUBJECT<<<"
T13107	C13180	FSNVSE	5	14	! ">>>\$TALK<<<"
T13108	C13190	FSNVSE	5	14	! ">>> ? <<"
T13109	C13200	FSNVSE	5	14	! ">>>\$INSERT<<<"
T13099	C13210	FSNVSE	5	14	! "INVALID DESCRIBE ARGUMENT PASSED"
C13110	C13500	D560	SRFN	TLCF	outnup ! \$describe text
T13110	C13010	FSNVSE	358	10	! "THIS PARAMETER IS USED TO OBTAIN ... "
C13120	C13500	D560	SRFN	TLCF	outnup ! \$command text
T13120	C13010	FSNVSE	159	17	! "THIS PARAMETER WILL CAUSE A ... "
C13130	C13500	D560	SRFN	TLCF	outnup ! \$delete text
T13130	C13010	FSNVSE	110	14	! "THIS PARAMETER WILL CAUSE THE ... "
C13140	C13500	D560	SRFN	TLCF	outnup ! \$end text
T13140	C13010	FSNVSE	357	10	! "THIS PARAMETER WILL CAUSE A ... "
C13150	C13500	D560	SRFN	TLCF	outnup ! \$list text
T13150	C13010	FSNVSE	214	10	! "\$LIST AS THE FIRST CHARACTERS ... "



C13160	C13500	D560	SRFN	TLCF	outnup ! \$select text
T13160	C13010	FSNVSE	256	10	! "THIS VERB ALLOWS A PARTICIPANT ... "
C13170	C13500	D560	SRFN	TLCF	outnup ! \$subject text
T13170	C13010	FSNVSE	300	20	! "THIS VERB ALLOWS A PARTICIPANT ... "
C13180	C13500	D560	SRFN	TLCF	outnup ! \$stalk text
T13180	C13010	FSNVSE	233	7	! "THIS PARAMETER WILL CAUSE A ... "
C13190	C13500	D13190	SRFN	TLCF	outnup ! ? text
T13190	C13010	FSNVSE	143	15	! "THE QUESTION MARK IS USED ... "
C13200	C13500	D560	SRFN	TLCF	outnup ! \$insert text
T13200	C13010	FSNVSE	636	27	! "THIS PARAMETER ALLOWS PREVIOUSLY ... "
C13210	C13500	D560	SRFN	TLCF	outnup ! ? or bad describe argument
T13210	C13010	FSNVSE	855	19	! "THIS TERMINAL IS CURRENTLY ... "
C13300	C13500	D13300	SNVN	TLCF	outnup ! \$delete command
T13300	C13300	FNNVSE	0	0	!<CR><NL>MESSAGE DELETED BY PARTICIPANT
T13300	C13300	VSNVSE	0	0	!<CR><NL>MESSAGE DELETED BY PARTICIPANT
T12000	C12010	FSNVSE	2	9	! "COMMAND ?" (command mode entered)
T11500	C11510	VSNVSE	0	0	! "LISTEN MODE ENTERED IN ... "
C13320	C13500	D13320	SNFN	TLCF	outnup ! \$send command
T13320	C13330	FSNVSE	0	18	! "MESSAGE ACCEPTED"
C13330	C13500	D13330	SRVN	TLCF	outnup ! Leave talk mode
T11500	C11510	VSNVSE	0	0	! "LISTEN MODE ENTERED IN ... "
T12000	C12010	VSNVSE	0	0	! "COMMAND ?"
C13340	C13500	D13340	SRVN	TLCF	outnup ! \$list command
T13340	C13050	FSNVSE	2	24	! "NO TEXT HAS BEEN ENTERED"
T13010	C13050	VSNVSE	0	0	! ">" (talk mode prompt)
C13360	C13500	D13360	SNVN	TLCF	outnup ! \$select command
T13360	C13010	FSNVSE	0	55	! "... NO PARTICIPANT NAMES"
T13010	C13050	VSNVSE	0	0	! ">" (talk mode prompt)
T13011	C13050	VSNVSE	0	0	! "-" (alternate talk mode prompt)
C13380	C13500	D13000	SNFN	TLCF	outnup ! \$subject command
T13380	C13020	FSNVSE	0	3	! "-"
T13381	C13050	FSNVSE	0	48	! "INVALID SUBJECT REQUEST--..."
C13410	C13500	D13410	SNFN	TLCF	outnup ! \$stalk command
T13320	C13420	FSNVSE	0	18	! "MESSAGE ACCEPTED"
T13430	C13410	FNNVSE	0	0	!CONTINUING IN TALK MODE<SP>
T13010	C13050	FNNVSE	0	0	!<CR><NL>>
T13010	C13050	VSNVSE	0	0	!<CR><NL>> preceded by message

C13420	C13500	D13420	SNVN	TLCF	outnup ! message number assigned
T13420	C13430	VSNVSE	0	0	! "MESSAGE NUMBER ... "
C13430	C13500	D13430	SNFN	TLCF	outnup ! return to talk mode
T13430	C13010	FNNVSE	0	0	! "CONTINUING IN TALK MODE"
C13500	C13500	D13300	SNVN	TLCF	outnup ! user entered a break
T13300	C13510	FNNVSE	0	0	!<CR><NL>MESSAGE DELETED BY PARTICIPANT
T13300	C13510	VSNVSE	0	0	!<CR><NL>MESSAGE DELETED BY PARTICIPANT
T12000	C12010	FSNVSE	0	9	! "COMMAND ?" (command mode entered)
T11500	C11510	VSNVSE	0	0	! "LISTEN MODE ENTERED IN ... "
C13510	C13500	D40	UNVN	TLCF	outnup ! user enters carriage return
T11040	C11500	VNNVUE	0	0	! (anything)

### A.5.2 Source Halting Delimiter File (GHDEL)

D1        |nl|^  
D20       -^  
D30       | |^  
D40       |cr|^  
D50       ZMA|cr|^  
          KED|cr|^  
D110      ?|cr||nl|^  
D130      ?|cr||nl|^  
          code^^  
D150      ?^  
          code^^  
D501      ?    ^^  
D502      ?    ^  
          ERS^^  
D503      6    ^^  
D504      L    ^^  
D506      y|cr||nl|^  
D509      6    ^  
          ?    ^  
          M ?^^  
D510      ?    ^  
          CODE^^  
D520      -^  
          DY^  
          ?    ^  
          USY ^^  
D530      ?    ^  
          DY^  
          FILE^^

0539 ERS^  
 VED-^  
 T CHARACTER^  
 LONG^  
 TION^  
 E SAVE^^

0540 ? ^  
 DY^  
 ERS^  
 W-^  
 E SAVE^^

0550 |cr||nl|^~

0560 |cr||nl|^~

0570 |cr||nl|INVALID COMMAND AT THIS LEVEL ^  
 |cr||nl|^  
 TED ^^

0580 ? ^  
 E|cr||nl|^  
 R |cr||nl|^  
 LEVEL ^  
 R FILE ^^

0581 ? ^^

0598 R FILE ^  
 R ^  
 - ^  
 E ^  
 R |cr||nl|^~

0599 ? ^  
 ^^

0611 - ^^

0990 |nl||nl|^  
 CP^^

0995 |nl||nl|^~

0999 |nl|^  
 |cr|^  
 | || |^^

01000 ? ^^

01030 | || || |^^  
 01050 ? ^  
 INPUT CHARACTER^  
 ' TOO LONG^^  
 01200 RWM|cr|^  
 STATED^^  
 01240 ? ^  
 INPUT CHARACTER^  
 ' TOO LONG^  
 E PERMISSION^^  
 01300 ? ^  
 EXIST^  
 FUL.^  
 INPUT CHARACTER^  
 ' TOO LONG^  
 STATED^^  
 06000 E|nl||cr|^  
 06120 | ||nl||cr|^  
 E|nl||cr|^  
 Y|nl||cr|^  
 06140 | ||nl||cr|^  
 D|nl||cr|^  
 06160 D|nl||cr|^  
 I|nl||cr|^  
 06260 | || || ||nl||cr|^  
 06340 | ||nl||cr|^  
 R|nl||cr|^  
 07000 y|cr||nl|^  
 cter^  
 NED|cr||nl|^  
 M ?^^  
 09020 Y^^  
 09030 |cr||nl|\*^^  
 09040 .^^  
 09045 |cr||nl|=^^

D9050 .  
 |cr||nl|\*  
 D9060 |cr||nl|=  
 |cr||nl|\*  
 D9061 \*  
 ?  
 D9063 E|cr||nl|  
 D9064 ?  
 D9065 TEM  
 D9070 |cr||nl|\*  
 |cr||nl|  
 ?  
 D9080 ?  
 LESS  
 |cr||cr||nl|  
 D9095 of 74  
 of SS  
 found  
 |cr||nl|  
 D9096 t|cr||nl|  
 \*  
 D9097 -  
 D9100 ?  
 \*  
 busy  
 found  
 D9120 |cr||nl|\*  
 |cr||nl|  
 |bell||bell||bell||bell||bell||bell|  
 TEM  
 ALLY  
 D11000 | |cr||nl|  
 CODE  
 D11010 JOIN?  
 D11030 JOIN?

CP^^

D11050 NAME? ^  
SESSION^^

D11070 ZMA|cr|^  
NIZED^  
DENIED ^^

D11090 COPY ^  
GRANTED ^  
DENIED ^  
ZMA|cr|^^

D11110 NAME ^  
CARD ^  
CARD--^^

D11130 ^^

D11140 |cr|^^

D11180 |cr||nl|^^

D11190 CONFERENCE ^^

D11510 |cr||nl|^  
CONFERENCE ^  
|ff|^^

D12000 COMMAND ?^^

D12040 WAITING^  
MESSAGE NUMBER^^

D12050 |cr||nl||ff|^  
|cr||nl|^  
COMMAND ?^^

D12060 AGE ^  
ENT ^^

D12070 -^  
SENT ^  
IGNORED ^  
COMMAND ?^^

D12100 COMMAND ?^  
CHANGE? |cr||nl|^  
LIMIT IS-- ^^

D12120 ? |cr||nl|^  
 ? |cr||nl|^  
 ? |cr||nl|^  
 COMMAND ?^^  
  
 D12200 MAND^  
 LEGE^  
 ?^^  
  
 D12210 >|cr||nl|^  
  
 D12230 TIVITY^^  
  
 D12240 ONLY^^  
  
 D12250 VALID ^  
 HIGH ^  
 CHANGE?^  
 COMMAND ?^^  
  
 D12270 COPY ^  
 NAME ^  
 CHANGE? ^  
 CARD |cr||nl|^  
 NO)?^  
 COMMAND ?^^  
  
 D12350 FUL ^^  
  
 D12360 EST ^  
 |ff|^  
  
 D12450 |nl|^  
 ?^^  
  
 D12500 |cr||nl|^  
 <|cr||nl|^  
 PASSED^^  
  
 D13000 |yy|^  
  
 D13010 >^  
 ^  
 ?^^  
  
 D13100 >|cr||nl|^  
 |cr||nl|^  
 PASSED^^  
  
 D13190 |cr||nl|^



UNCTION.        ^^

D13300 PANT ^  
      |cr||nl|^  
MAND ?^^

D13320 TED^^

D13330 |yy| |cr||nl|^  
COMMAND ?^^

D13340 ENTERED^  
      |cr||nl|>^  
      |cr||nl|-^  
      |cr|^^

D13360 NAMES|cr||nl|^  
      |cr||nl|>^  
      |cr||nl|-^^

D13400 |cr||nl|-^^

D13410 |cr||nl|>^  
MODE ^  
ACCEPTED^^

D13420 |cr||nl|^^

D13430 MODE ^^

D13440 |cr||nl||ff|^^

### A.5.3 Source Text String File (GTEXT)

Long text string lines have been "wrapped around" onto the next line in order to fit onto the page. This wrapping is indicated by the characters "->" at the end of a line.

```
T1      ^
T2      |cr||nl|1316400|cr||nl|^
T10     |cr||nl|PROGRAM NAME -^
T20     tss|cr|^
T21     tlc|cr|^
T30     |cr||nl|TERMINAL ^
T40     |cr||nl|^
T50     USERID$PASSWORD?|cr||nl||cr||nl|
        *%<%#>?*$%#?AM*5*48#N&#|cr|
        TXOGWBMWEPVRMCBQNKMGAKZMA|cr|^
T51     LOCKED|cr|^
T60     /^
T61     $^
T90     |cr||nl|IDENT?|cr||nl|^
T100    .^
T110    |cr||nl|CLASSIFICATION OF YOUR OUTPUT?|cr||nl|^
T111    |cr||nl|classification of your output?|cr||nl|^
T120    zzz|cr|^
T121    uzz|cr|^
T122    ufo|cr|^
T123    czz|cr|^
```

T124 szz|cr|^  
 T125 swp|cr|^  
 T130 |cr||nl|CLASSIFICATION OF FILES YOU WILL CREATE?|cr||nl|^  
 T131 |cr||nl|classification of files you will create?|cr||nl|^  
 T132 |cr||nl|ILLEGAL CLASSIFICATION CODE^  
 T133 |cr||nl|illegal classification code^  
 T150 |cr||nl|SYSTEM ?^  
 T151 |nl||cr||cr||nl|SYSTEM ?^  
 T160 acce|cr|^  
 T161 acces|cr|^  
 T162 access|cr|^  
 T163 list ^  
 T164 sios|cr|^  
 T165 tcon|cr|^  
 T166 wwdm|cr|^  
 T167 wwdms|cr|^  
 T168 bye|cr|^  
 T169 listl ^  
 T170 list|cr|^  
 T171 listl|cr|^  
 T172 wwdm n|cr|^  
 T173 wwdms n|cr|^  
 T174 wwdm new|cr|^  
 T175 wwdms new|cr|^  
 T991 |cr||nl| LINE TERMINATED CP^

T995 |cr||nl|\*\*on at ^  
 T1000 |cr||nl||nl| FUNCTION? ^  
 T1020 mf|cr|^  
 T1021 fm|cr|^  
 T1022 done|cr|^  
 T1023 |cr|^  
 T1030 |cr||nl| CATALOG STRUCTURE TO WORKING LEVEL?|cr||nl| ^  
 T1040 ^  
 T1050 |cr||nl| FILE TO BE MODIFIED? ^  
 T1051 |cr||nl||nl|ERR-^  
 T1060 /^  
 T1070 |cr||nl| NEW NAME? ^  
 T1090 |cr||nl| NEW MAX SIZE IN LLINKS? ^  
 T1200 |cr||nl|NEW PASSWORD?|cr||nl|  
 a%&<%#>?#a\$%&#|cr|  
 TXOGWBMWEPVRWM|cr|^  
 T1201 |cr||nl||nl|ERR--MAX SIZE ILLEGALLY STATED^  
 T1220 |cr||nl| GENERAL PERMISSIONS? ^  
 T1230 read|cr|^  
 T1231 r|cr|^  
 T1232 execute|cr|^  
 T1233 e|cr|^  
 T1234 write|cr|^  
 T1235 w|cr|^  
 T1236 append|cr|^  
 T1237 a|cr|^

T1238 modify|cr|^  
 T1239 m|cr|^  
 T1240 |cr||nl| SPECIFIC PERMISSIONS? ^  
 T1250 read/^  
 T1251 r/^  
 T1252 write/^  
 T1253 w/^  
 T1254 append/^  
 T1255 a/^  
 T1256 execute/^  
 T1257 e/^  
 T1258 modify/^  
 T1259 m/^  
 T1300 |cr||nl|MORE? ^  
 T1301 |cr||nl| SUCCESSFUL.^  
 T1302 |cr||nl||nl|ERR-PERMISSIONS ILLEGALLY STATED^  
 T1303 |cr||nl||nl|REQUEST DENIED-^  
 T1320 |cr||nl||nl| FILE TO BE MODIFIED? ^  
 T2000 |cr||nl|TCON VERSION 6.3, JUNE 1976 ^  
 T2001 |cr||nl|FUNCTIONAL AREA ? ^  
 T2002 |cr||nl|TCON VERSION 6.3, JUNE 1976  
 |cr||nl|FUNCTIONAL AREA ? ^  
 T2020 |cr||nl|NO TEMPORARY FILE SPACE CODE^  
 T2030 fs|cr|^  
 T2040 dem|cr|^  
 T2050 |cr||nl|^

T2060 0 ^  
 T2061 1 ^  
 T2062 2 ^  
 T2063 3 ^  
 T2064 4 ^  
 T2065 5 ^  
 T2066 6 ^  
 T2067 7 ^  
 T2068 8 ^  
 T2069 9 ^  
 T2100 |cr|^  
 T2110 |cr||nl|OLD OR NEW-^  
 T2120 |cr||nl|FUNCTIONAL AREA BUSY ^  
 T2130 new|cr|^  
 T2140 n|cr|^  
 T2160 o ^  
 T2161 ol ^  
 T2162 old ^  
 T2180 same|cr|^  
 T2181 s|cr|^  
 T2190 |cr||nl|READY^  
 T2195 -- NONEXISTENT^  
 T2196 ' DOES NOT EXIST--USE SAVE^  
 T2200 |cr||nl||nl||cr||nl|TRANSACTION TYPE ? ^  
 T2201 |cr||nl| TRANSACTION TYPE ? ^

T2202 |cr||nl|TRANSACTION TYPE ? ^  
 T2205 |cr||nl||nl||cr||nl|INVALID TRANSACTION TYPE ON RETRIEVE  
 |cr||nl||cr||nl|ENTER TRANSACTION TYPE ^  
 T2210 |cr||nl|YOU PRESENTLY DO NOT HAVE A CURRENT FILE^  
 T2211 |cr||nl|FILE NAME? ^  
 T2220 INCORRECT CAT/FILE DESCRIPTION^  
 T2221 |cr||nl||nl|ERR-FILE NAME >8 CHARACTERS^  
 T2222 |cr||nl|ERR-FILE NAME >8 CHARACTERS^  
 T2223 IS NOT A LEGAL INPUT CHARACTER^  
 T2224 IS AN ILLEGAL INPUT CHARACTER^  
 T2225 TOO LONG^  
 T2230 |cr||nl||nl||cr||nl|INVALID TRANSACTION TYPE -RETYPE- ^  
 T2240 NOT IN CURR FILE ^  
 T2250 |cr||nl|ILLEGAL PARAMETER ^  
 T2260 FILE PRESENTLY BUSY^  
 T2270 |cr||nl|DATA SAVED-^  
 T2280 |cr||nl||cr||nl||dc3||cr||nl|ready|cr||nl|^  
 T2281 |cr||nl|ready|cr||nl|^  
 T2290 a|cr|^  
 T2300 a a|cr|^  
 T2310 aa1|cr|^  
 T2320 d|cr|^  
 T2330 e|cr|^  
 T2340 h|cr|^  
 T2350 j|cr|^  
 T2360 k|cr|^

T2370 ka1|cr|^  
T2380 l|cr|^  
T2390 m|cr|^  
T2400 n|cr|^  
T2410 p|cr|^  
T2420 r-1|cr|^  
T2430 r-2|cr|^  
T2440 r11|cr|^  
T2450 r12|cr|^  
T2460 aa4|cr|^  
T2470 ka2|cr|^  
T2480 t|cr|^  
T2490 v|cr|^  
T2500 x|cr|^  
T2510 z a|cr|^  
T2512 jsb|cr|^  
T2514 j b|cr|^  
T2516 n a|cr|^  
T2518 n b|cr|^  
T2519 xxx|cr|^  
T2520 done|cr|^  
T2540 dele^  
T2541 feed^  
T2542 next^  
T2543 nofe^



T2544 pass^  
T2545 rewi^  
T2546 skip^  
T2547 help^  
T2548 voca^  
T2550 remove clearfiles|cr|^  
T2555 remov clearfiles|cr|^  
T2560 remo clearfiles|cr|^  
T2580 rese^  
T2600 retr^  
T2610 list|cr|^  
T2611 list ^  
T2612 listl|cr|^  
T2613 listl ^  
T2630 resave|cr|^  
T2631 resa|cr|^  
T2632 resave ^  
T2633 resa ^  
T2640 old|cr|^  
T2645 o|cr|^  
T2890 tn|cr|^  
T2891 tn^  
T2910 [^  
T2940 forward^  
T2941 f^

T2960 backward^  
 T2961 b^  
 T2980 get ^  
 T2981 g ^  
 T3000 test|cr|^  
 T3001 t|cr|^  
 T3010 \*|cr|^  
 T3011 system|cr|^  
 T3012 syst|cr|^  
 T3020 xmit|cr|^  
 T3021 x|cr|^  
 T3030 bye|cr|^  
 T3040 d|cr|^  
 T3050 DATA TRUNCATED ^  
 T3060 |cr||nl|INVALID COMMAND AT THIS LEVEL ^  
 T3070 END OF TRANSACTION |cr||nl|^  
 T3080 MORE - - - -|cr||nl|^  
 T3090 |cr||nl||nl||cr||nl|TRANSACTION SENT. NEXT? ^  
 T3110 |cr||nl|OLD FILE? ^  
 T3120 |cr||nl|INVALID COMMAND AT THIS LEVEL |cr||nl|^  
 T3130 |cr||nl||nl||cr||nl|INVALID TRANSACTION TYPE -RETYPE- ^  
 T3300 hd|cr|^  
 T3310 as|cr|^  
 T3320 am|cr|^  
 T3330 av|cr|^



DATA	NAME	ERROR MESSAGE cr  nl
01[---	A-A-CARD-SEQ cr  nl	
02[-	A-A-SECURITY-CL cr  nl	
03[-	A-A-TRAN-TYPE cr  nl	
04[A A	A-A-CARD-TYPE cr  nl	
05[-----	A-A-UIC cr  nl	
06[-----	A-A-SRC cr  nl	
07[-----	TAPFOR-CODE cr  nl	
08[-	A-A-PACING cr  nl	
09[--	A-A-RDYCD cr  nl	
10[-	A-A-TPSNCD cr  nl	
11[-----	A-A-SBRPT2 cr  nl	
12[----	A-A-ROBCO1 cr  nl	
13[-	A-A-MSYSACD cr  nl	
14[---	A-A-ROBCO cr  nl	
15[-----	A-A-DAMPL cr  nl	
16[----	A-A-SEQKEY cr  nl	
17[-	A-A-IFREQ cr  nl	
18[-	A-A-ARR cr  nl	
19[--	A-A-ARGO cr  nl	
20[-	A-A-ARMY cr  nl	
21[-----	A-A-ORGUIC cr  nl	
22[FS	A-A-RPTYP cr  nl	
23[---	A-A-RPTNR cr  nl  cr  nl ~	

T2680 |cr||nl|

DATA	NAME	TRANSACTION SEQUENCE NUMBER  cr  nl	ERROR MESSAGE cr  nl
01[---	AA1-CARD-SEQ cr  nl		
02[-	AA1-SECURITY-CL cr  nl		
03[-	AA1-TRAN-TYPE cr  nl		
04[AA1	AA1-CARD-TYPE cr  nl		
05[-----	AA1-UIC cr  nl		
06[-	AA1-LIC cr  nl		
07[-----	AA1-EDATE cr  nl		
08[-----	AA1-TPSN15 cr  nl		
09[--	AA1-TPSN-EL-SEQ cr  nl		
10[-	AA1-OESTS cr  nl		
11[--	AA1-STATC cr  nl		
12[-	AA1-FUAC cr  nl		
13[--	AA1-ASGMT cr  nl		
14[-----	AA1-ZIP cr  nl		
15[----	AA1-APO cr  nl		
16[--	AA1-PPA cr  nl		
17[-----	AA1-MAC cr  nl		
18[----	AA1-PLGEO cr  nl		
19[-----	AA1-ORGUIC cr  nl		
20[FS	AA1-RPTYP cr  nl		
21[---	AA1-RPTNR cr  nl  cr  nl ~		

T2690 |cr||nl|  
|cr||nl|TNC000010 TRANSACTION SEQUENCE NUMBER |cr||nl|  
DATA NAME ERROR MESSAGE|cr||nl|  
U1C--- D-CARD-SEQ|cr||nl|  
U2C- D-SECURITY-CL|cr||nl|  
U3C- D-TRAN-TYPE|cr||nl|  
U4Cd D-CARD-TYPE|cr||nl|  
U5C----- D-UIC|cr||nl|  
U6C- D-CSERV|cr||nl|  
U7C----- D-OPCON|cr||nl|  
U8C----- D-ADCON|cr||nl|  
U9C---- D-HOGEO|cr||nl|  
U10C---- D-PRGEO|cr||nl|  
U11C----- D-EMBRK|cr||nl|  
U12C-- D-ACTIV|cr||nl|  
U13C- D-DFCON|cr||nl|  
U14C- D-NUCIN|cr||nl|  
U15C- D-PCTEF|cr||nl|  
U16C----- D-ORGUIC|cr||nl|  
U17CFS D-RPTYP|cr||nl|  
U18C--- D-RPTNR|cr||nl||cr||nl|~

T2700 |cr||nl|  
|cr||nl|TNC000010 TRANSACTION SEQUENCE NUMBER |cr||nl|  
DATA NAME ERROR MESSAGE|cr||nl|  
O1C--- E-CARD-SEQ|cr||nl|  
O2C- E-SECURITY-CL|cr||nl|  
U3CE E-CARD-TYPE|cr||nl|  
U4C----- E-ORGUIC|cr||nl|  
O5CFS E-RPTYP|cr||nl|  
U6C--- E-RPTNR|cr||nl||cr||nl|~

T2710 |cr||nl|  
|cr||nl|TNC000010 TRANSACTION SEQUENCE NUMBER |cr||nl|  
DATA NAME ERROR MESSAGE|cr||nl|  
U1C001 H-CARD-SEQ|cr||nl|  
U2C- H-SECURITY-CL|cr||nl|  
U3CH H-CARD-TYPE|cr||nl|  
U4C-- H-CARD-DAY|cr||nl|  
U5C-- H-CARD-HR|cr||nl|  
U6C-- H-CARD-MIN|cr||nl|  
U7CZ H-ZULU|cr||nl|  
U8C--- H-CARD-MONTH|cr||nl|  
U9C-- H-CARD-YEAR|cr||nl|  
U10C- H-REAL-EXERCISE|cr||nl|  
U11C----- H-ORGUIC|cr||nl|  
U12CFS H-RPTYP|cr||nl|  
U13C--- H-RPTNR|cr||nl||cr||nl|~

T2720 |cr||nl|

DATA	NAME	TRANSACTION SEQUENCE NUMBER	ERROR MESSAGE
01[---	J-CARD-SEQ		
02[-	J-SECURITY-CL		
03[-	J-TRAN-TYPE		
04[CJ	J-CARD-TYPE		
05[-----	J-UIC		
06[-	J-TPERS		
07[-----	J-PEGE0		
08[-----	J-STRUC		
09[-----	J-AUTH		
10[-----	J-ASGD		
11[-----	J-POSTR		
12[-----	J-PICDA		
13[-----	J-DEPS		
14[-----	J-CASPW		
15[-----	J-CCASP		
16[-----	J-ORGUIC		
17[FS	J-RPTYP		
18[---	J-RPTNR		

T2730

DATA	NAME	TRANSACTION SEQUENCE NUMBER	ERROR MESSAGE
01[---	CARD-SEQ		
02[-	SECURITY-CL		
03[-	TRAN-TYPE		
04[---	CARD-TYPE		
05[-----	UIC		
06[-	READY		
07[-	REASN		
08[-	PRRAT		
09[---	PRRES		
10[-	ESRAT		
11[---	ESRES		
12[-	ERRAT		
13[---	ERRES		
14[-	TRRAT		
15[---	TRRES		
16[---	SECRN		
17[---	TERRN		
18[-	CARAT		
19[-----	CADAT		
20[-	LIM		
21[-	RLIM		
22[-----	RICDA		
23[-----	ORGUIC		
24[FS	RPTYP		
25[---	RPTNR		

T2740		TRANSACTION SEQUENCE NUMBER	
DATA	NAME	ERROR MESSAGE	
01[---	CARD-SEQ		
02[-	SECURITY-CS		
03[-	TRANS-TYPE		
04[---	CARD-TYPE		
05[-----	UIC		
06[---	PSPER		
07[-	MSPER		
08[-	SGPER		
09[-	APERT		
10[-	DEPST		
11[---	EHRDN		
12[---	EHRD1		
13[---	EHRD2		
14[---	EHRD3		
15[---	EHRD4		
16[-	PILFIL		
17[-	PERRY		
18[-	PI-ES		
19[-	ER		
20[-	PI-ER		
21[-	TWRC1		
22[-	ITAVS		
23[-	ITBMM		
24[-	ITAFT		
25[-	ITAEM		
26[-	ITAQL		
27[-	ITATF		
28[-	ITAFI		
29[-	ITATA		
30[-	ITATM		
31[-	READ2		
32[-	ALO		
33[-----	RICD1		
34[-	PUID		
35[-----	ORGUIC		
36[FS	RPTYP		
37[---	RPTNR		

T2750		TRANSACTION SEQUENCE NUMBER	
DATA	NAME	ERROR MESSAGE	
01[---	L-CARD-SEQ		
02[-	L-SECURITY-CL		
03[-	L-TRAN-TYPE		
04[CL	L-CARD-TYPE		
05[-----	L-UIC		
06[-----	L-MEQPT		

07[---	L-MEPSA cr  nl
08[---	L-MEPSD cr  nl
09[---	L-MEORD cr  nl
10[---	L-MEORN cr  nl
11[-----	L-MEREC cr  nl
12[-----	L-ORGUIC cr  nl
13[FS	L-RPTYP cr  nl
14[---	L-RPTNR cr  nl  cr  nl ^

T2760	cr  nl		
	cr  nl TN[000010	TRANSACTION SEQUENCE NUMBER	cr  nl
	DATA	NAME	ERROR MESSAGE cr  nl
01[---		M-CARD-SEQ cr  nl	
02[---		M-SECURITY-CL cr  nl	
03[---		M-TRAN-TYPE cr  nl	
04[CM		M-CARD-TYPE cr  nl	
05[-----		M-UIC cr  nl	
06[-----		M-MEQPT cr  nl	
07[-----		M-TEGEO cr  nl	
08[---		M-MEPSD cr  nl	
09[---		M-MEORD cr  nl	
10[---		M-MEORN cr  nl	
11[-----		M-MEREC cr  nl	
12[-----		M-ORGUIC cr  nl	
13[FS		M-RPTYP cr  nl	
14[---		M-RPTNR cr  nl  cr  nl ^	

T2770	cr  nl		
	cr  nl TN[000010	TRANSACTION SEQUENCE NUMBER	cr  nl
	DATA	NAME	ERROR MESSAGE cr  nl
01[---		N-CARD-SEQ cr  nl	
02[---		N-SECURITY-CL cr  nl	
03[---		N-TRAN-TYPE cr  nl	
04[CM		N-CARD-TYPE cr  nl	
05[-----		N-UIC cr  nl	
06[-----		N-PIN cr  nl	
07[-----		N-FRN cr  nl	
08[---		N-PLEAC cr  nl	
09[---		N-DDP cr  nl	
10[-----		N-DDPRD cr  nl	
11[-----		N-MDT cr  nl	
12[-----		N-PUTCV cr  nl	
13[-----		N-ORGUIC cr  nl	
14[FS		N-RPTYP cr  nl	
15[---		N-RPTNR cr  nl  cr  nl ^	

T2780	cr  nl		
	cr  nl TN[000010	TRANSACTION SEQUENCE NUMBER	cr  nl
	DATA	NAME	ERROR MESSAGE cr  nl
01[---		P-CARD-SEQ cr  nl	



02[-	P-SECURITY-CL cr  nl
03[-	P-TRAN-TYPE cr  nl
04[P	P-CARD-TYPE cr  nl
05[-----	P-UIC cr  nl
06[-----	P-PIN cr  nl
07[-----	P-PEQPT cr  nl
08[-----	P-TPGEO cr  nl
09[--	P-ALTYP cr  nl
10[---	P-NUMRR cr  nl
11[---	P-NUMEA cr  nl
12[-----	P-ALRET cr  nl
13[-----	P-ORGUIC cr  nl
14[FS	P-RPTYP cr  nl
15[---	P-RPTNR cr  nl  cr  nl ^

T2790	cr  nl			
	cr  nl TNE000010	TRANSACTION SEQUENCE NUMBER	cr  nl	
	DATA	NAME	ERROR MESSAGE c->	
	r  nl			
01[---		R-1-CARD-SEQ cr  nl		
02[-		R-1-SECURITY-CL cr  nl		
03[-		R-1-TRAN-TYPE cr  nl		
04[---		R-1-CARD-TYPE cr  nl		
05[-----		R-1-UIC cr  nl		
06[1		R-1-RMK-SEQ-NUM cr  nl		
07[-		R-1-NUM-OF-CARDS cr  nl		
08[-----		R-1-RMK-LABEL cr  nl		
09[-----		R-1-SEC-CONTROL cr  nl		
10[---		R-1-RMK-DATE cr  nl		
11[-----		R-1-RMAE cr  nl		
12[-----		R-1-RMK-NARRATIVE cr  nl		
13[-----		R-1-ORGUIC cr  nl		
14[FS		R-1-RPTYP cr  nl		
15[---		R-1-RPTNR cr  nl  cr  nl ^		

T2800	cr  nl			
	cr  nl TNE000010	TRANSACTION SEQUENCE NUMBER	cr  nl	
	DATA	NAME	->	
	ERROR MESSAGE cr  nl			
01[---		R-2-CARD-->		
SEQ cr  nl				
02[-		R-2-SECUR-->		
ITY-CL cr  nl				
03[-		R-2-TRAN-->		
TYPE cr  nl				
04[---		R-2-CARD-->		
TYPE cr  nl				
05[-----		R-2-UIC c->		
r  nl				
06[-		R-2-RMK-S->		

EQ-NUM cr  nl	
07[-	R-2-NUM-0->
F-RMKS cr  nl	
08[-----	R-2-RMK-N->
ARRATIVE cr  nl	
09[-----	R-2-ORGUI->
C cr  nl	
10[FS	R-2-RPTYP->
cr  nl	
11[---	R-2-RPTNR->
cr  nl  cr  nl ^	

T2810	cr  nl		
	cr  nl TN[000010	TRANSACTION SEQUENCE NUMBER	cr  nl
	DATA		NAME ->
	ERROR MESSAGE cr  nl		
	01[---		R-11-CARD-SEQ cr  ->
	nl		
	02[-		R-11-SECURITY-CL c->
	r  nl		
	03[-		R-11-TRAN-TYPE cr ->
	nl		
	04[---		R-11-CARD-TYPE cr ->
	nl		
	05[-----		R-11-UIC cr  nl
	U6[1		R-11-RMK-SEQ-NUM c->
	r  nl		
	07[-		R-11-NUM-OF-RMKS c->
	r  nl		
	08[-----		R-11-RMK-LABEL cr ->
	nl		
	09[-----		R-11-RMK-DATE cr  ->
	nl		
	10[-----		R-11-RMK-NARRITI c->
	r  nl		
	11[-----		R-11-ORGUIC cr  nl->
	12[FS		R-11-RPTYP cr  nl
	13[---		R-11-RPTNR cr  nl ->
	cr  nl ^		

T2820	cr  nl		
	cr  nl TN[000010	TRANSACTION SEQUENCE NUMBER	cr  nl
	DATA		NAME ->
	ERROR MESSAGE cr  nl		
	U1[---		R-12-CARD->
	-SEQ cr  nl		
	U2[-		R-12-SECJ->
	RITY-CL cr  nl		
	U3[-		R-12-TRAN->

-TYPE cr  nl	
U4C---	R-12-CARD->
-TYPE cr  nl	
U5C-----	R-12-UIC ->
cr  nl	
U6C-	R-12-RMK-->
SEQ cr  nl	
U7C-	R-12-NUM-->
RMKS cr  nl	
U8C-----	R-12-RMK-->
NARRIT cr  nl	
U9C-----	R-12-ORGU->
IC cr  nl	
10CFS	R-12-RPTY->
P cr  nl	
11C---	R-12-KPTN->
R cr  nl  cr  nl ^	

T2830	cr  nl		
	cr  nl TNC000010	TRANSACTION SEQUENCE NUMBER	cr  nl
	DATA	NAME	ERROR MESSAGE cr  nl
	U1C---	AA4-SEQ cr  nl	
	U2C-	AA4-SECURITY-CL cr  nl	
	U3C-	AA4-TRANS-TYPE cr  nl	
	U4CAA4	AA4-CARD-TYPE cr  nl	
	U5C-----	AA4-UIC-CONTROL cr  nl	
	U6C--	AA4-ASGMT cr  nl	
	U7C-----	AA4-TPSN15 cr  nl	
	U8C--	AA4-TPSN-EL-SEQ cr  nl	
	U9C-----	AA4-SRC cr  nl	
	10C-	AA4-OESTS cr  nl	
	11C--	AA4-STATC cr  nl	
	12C-----	AA4-MAC cr  nl	
	13C-----	AA4-POAD cr  nl	
	14C-----	AA4-PIDD cr  nl	
	15CWOZUFF	AA4-ORGUIC cr  nl	
	16CFS	AA4-RPTYP cr  nl	
	17C---	AA4-RPTNBR cr  nl  cr  nl ^	

T2840	cr  nl		
	cr  nl TNC000010	TRANSACTION SEQUENCE NUMBER	cr  nl
	DATA	NAME	ERROR MESSAGE cr  nl
	U1C---	CARD-SEQ cr  nl	
	U2C-	SECURITY-CL cr  nl	
	U3C-	TRAN-TYPE cr  nl	
	U4CKA2	CARD-TYPE cr  nl	
	U5C-----	UIC cr  nl	
	U6C---	PEOHN cr  nl	
	U7C---	PEOH1 cr  nl	
	U8C---	PEOH2 cr  nl	

09C---	PEOH3 cr  nl
10C---	PEOH4 cr  nl
11C--	PIEOH cr  nl
12C--	POMRR cr  nl
13C--	POMPI cr  nl
14C--	POMER cr  nl
15C--	PRIER cr  nl
16C-----	RICD2 cr  nl
17C-----	ORGUIC cr  nl
18CFS	RPTYP cr  nl
19C---	RPTNR cr  nl  cr  nl ^

T2850	cr  nl		
	cr  nl TN000010	TRANSACTION SEQUENCE NUMBER	cr  nl
	DATA	NAME	ERROR MESSAGE cr  nl
01C---		T-CARD-SEQ cr  nl	
02C-		T-SECURITY-CL cr  nl	
03C-		T-TRAN-TYPE cr  nl	
04CT		T-CARD-TYPE cr  nl	
05C-----		T-UIC cr  nl	
06C-----		T-TEQPT cr  nl	
07C----		T-MESEN cr  nl	
08C-		T-DECON cr  nl	
09C--		T-MECUS cr  nl	
10C-		T-AVCAT cr  nl	
11C-		T-RESND cr  nl	
12C-----		T-ERDTE cr  nl	
13C-		T-EXDAC cr  nl	
14C----		T-CPGEO cr  nl	
15C----		T-CFGEO cr  nl	
16C-----		T-EQDEP cr  nl	
17C-----		T-EQARR cr  nl	
18C-----		T-TPIN cr  nl	
19C-		T-TLEAC cr  nl	
20C--		T-TLEQE cr  nl	
21C-----		T-ORGUIC cr  nl	
22CFS		T-RPTYP cr  nl	
23C---		T-RPTNR cr  nl  cr  nl ^	

T2860	cr  nl		
	cr  nl TN000010	TRANSACTION SEQUENCE NUMBER	cr  nl
	DATA	NAME	ERROR MESSAGE cr  nl
01C---		V-CARD-SEQ cr  nl	
02C-		V-SECURITY-CL cr  nl	
03C-		V-TRAN-TYPE cr  nl	
04CV		V-CARD-TYPE cr  nl	
05C-----		V-UIC cr  nl	
06C----		V-ACGEO cr  nl	
07C--		V-ACITY cr  nl	
08C-----		V-ADATE cr  nl	

09[---- V-MDATE|cr||nl|  
 10[----- V-RDATE|cr||nl|  
 11[----- V-ORGUIC|cr||nl|  
 12[FS V-RPTYP|cr||nl|  
 13[--- V-RPTNR|cr||nl||cr||nl|^

T2870 |cr||nl|  
 |cr||nl|TNC000010 TRANSACTION SEQUENCE NUMBER |cr||nl|  
 DATA NAME ERROR MESSAGE|cr||nl|  
 01[--- X-CARD-SEQ|cr||nl|  
 02[- X-SECURITY-CL|cr||nl|  
 03[C X-TRAN-TYPE|cr||nl|  
 04[X X-CARD-TYPE|cr||nl|  
 05[----- X-UIC|cr||nl|  
 06[----- X-GCMD|cr||nl|  
 07[----- X-TDATE|cr||nl|  
 08[---- X-TRGEO|cr||nl|  
 09[----- X-DEPDT|cr||nl|  
 10[----- X-ARRDT|cr||nl|  
 11[----- X-RPTOR|cr||nl|  
 12[----- X-INTRST1|cr||nl|  
 13[----- X-INTRST2|cr||nl|  
 14[----- X-SBRPT|cr||nl|  
 15[--- X-ATACH|cr||nl|  
 16[----- X-ORGUIC|cr||nl|  
 17[FS X-RPTYP|cr||nl|  
 18[--- X-RPTNR|cr||nl||cr||nl|^

T2880 |cr||nl|  
 |cr||nl|TNC000010 TRANSACTION SEQUENCE NUMBER |cr||nl|  
 DATA NAME ER->  
 ROR MESSAGE|cr||nl|  
 01[--- Z-A-CARD-SEQ|cr||nl|  
 02[- Z-A-SECURITY-CL|cr||nl->  
 |  
 03[C Z-A-TRAN-TYPE|cr||nl|  
 04[Z A Z-A-CARD-TYPE|cr||nl|  
 05[----- Z-A-UIC|cr||nl|  
 06[--- Z-A-CARD-TYPE-IN|cr||n->  
 |  
 07[- Z-A-TRANS-CODE|cr||nl|  
 08[----- Z-A-ERROR-CODE|cr||nl|  
 09[--- Z-A-CARD-NUMBER|cr||nl->  
 |  
 10[--- Z-A-REPORT-NUMBER|cr||->  
 nl|  
 11[----- Z-A-ORGUIC|cr||nl|  
 12[FS Z-A-RPTYP|cr||nl|  
 13[--- Z-A-RPTNR|cr||nl||cr||->  
 nl|^

```

T3140 |cr||nl|
|cr||nl|TN000010 TRANSACTION SEQUENCE NUMBER |cr||nl|
DATA NAME ERROR MESSAGE|cr||nl|
01[--- J-B-CARD-SEQ|cr||nl|
02[- J-B-SECURITY-CL|cr||nl|
03[- J-B-TRAN-TYPE|cr||nl|
04[J B J-B-CARD-TYPE|cr||nl|
05[----- J-B-UIC|cr||nl|
06[- J-B-COMPO|cr||nl|
07[----- J-B-EDATE|cr||nl|
08[----- J-B-STRUCC|cr||nl|
09[----- J-B-STRUCW|cr||nl|
10[----- J-B-STRUCE|cr||nl|
11[----- J-B-AUTHO|cr||nl|
12[----- J-B-AUTHW|cr||nl|
13[----- J-B-AUTHE|cr||nl|
14[----- J-B-ORGUIC|cr||nl|
15[FS J-B-RPTYP|cr||nl|
16[--- J-B-RPTNR|cr||nl||cr||nl|^

```

```

T3150 |cr||nl|
|cr||nl|TN000010 TRANSACTION SEQUENCE NUMBER |cr||nl|
DATA NAME ERROR MESSAGE|cr||nl|
01[--- CARD-SEQ|cr||nl|
02[- SECURITY-CS|cr||nl|
03[- TRANS-TYPE|cr||nl|
04[J B CARD-TYPE|cr||nl|
05[----- UIC|cr||nl|
06[----- STRUCO|cr||nl|
07[----- STRUCW|cr||nl|
08[----- STRUCE|cr||nl|
09[----- AUTHO|cr||nl|
10[----- AUTHW|cr||nl|
11[----- AUTHE|cr||nl|
12[----- DAMPL|cr||nl|
13[--- ROBCO|cr||nl|
14[----- ORGUIC|cr||nl|
15[FS RPTYP|cr||nl|
16[--- RPTNR|cr||nl||cr||nl|^

```

```

T3160 |cr||nl|
|cr||nl|TN000010 TRANSACTION SEQUENCE NUMBER |cr||nl|
DATA NAME ERROR MESSAGE|cr||nl|
01[--- N-A-CARD-SEQ|cr||nl|
02[- N-A-SECURITY-CL|cr||nl|
03[- N-A-TRAN-TYPE|cr||nl|
04[N A N-A-CARD-TYPE|cr||nl|
05[----- N-A-UIC|cr||nl|
06[----- N-A-DISTR|cr||nl|
07[----- N-A-ORGUIC|cr||nl|

```

```

08[FS          N-A-RPTYP|cr||nl|
09[---          N-A-RPTNR|cr||nl||cr||nl|^

```

```

T3170 |cr||nl|
|cr||nl|TN000010      TRANSACTION SEQUENCE NUMBER |cr||nl|
      DATA          NAME          ERROR MESSAGE|cr||nl|
01[---          N-B-CARD-SEQ|cr||nl|
02[-            N-B-SECURITY-CL|cr||nl|
03[-            N-B-TRAN-TYPE|cr||nl|
04[ N B          N-B-CARD-TYPE|cr||nl|
05[-----      N-B-UIC|cr||nl|
06[-----      N-B-PIN|cr||nl|
07[-----      N-B-FRN|cr||nl|
08[-            N-B-RPTDES|cr||nl|
09[-            N-B-PLEAC|cr||nl|
10[-----      N-B-RTM|cr||nl|
11[---          N-B-RSNLA|cr||nl|
12[-----      N-B-PUTCV|cr||nl|
13[-----      N-B-POEGEO|cr||nl|
14[-----      N-B-PODGE0|cr||nl|
15[---          N-B-MODE|cr||nl|
16[-----      N-B-LAD|cr||nl|
17[-----      N-B-RTMDIR|cr||nl|
18[-            N-B-GCMDCODE|cr||nl|
19[-----      N-B-ORGUIC|cr||nl|
20[FS          N-B-RPTYP|cr||nl|
21[---          N-B-RPTNR|cr||nl||cr||nl|^

```

```

T3180 |cr||nl|
|cr||nl|TN000010      TRANSACTION SEQUENCE NUMBER |cr||nl|
      DATA          NAME          ERROR MESSAGE|cr||nl|
01[---          KA1-CARD-SEQ|cr||nl|
02[-            KA1-SECURITY-CL|cr||nl|
03[-            KA1-TRAN-TYPE|cr||nl|
04[KA1          KA1-CARD-TYPE|cr||nl|
05[-----      KA1-UIC|cr||nl|
06[-            KA1-EXLIM|cr||nl|
07[--           KA1-TWRC1|cr||nl|
08[---          KA1-PSPER|cr||nl|
09[--           KA1-MSPER|cr||nl|
10[--           KA1-APERT|cr||nl|
11[-----      KA1-DEPST|cr||nl|
12[-----      KA1-DUTYO|cr||nl|
13[---          KA1-DUTYW|cr||nl|
14[-----      KA1-DUTYE|cr||nl|
15[-            KA1-MSYSA|cr||nl|
16[-----      KA1-PERTL|cr||nl|
17[--           KA1-PERAB|cr||nl|
18[--           KA1-PERRS|cr||nl|
19[--           KA1-PERRE|cr||nl|

```

20C---	KA1-EHRDN cr  nl
21C---	KA1-EHRD1 cr  nl
22C---	KA1-EHRD2 cr  nl
23C---	KA1-EHRD3 cr  nl
24C---	KA1-EHRD4 cr  nl
25C-----	KA1-ORGUIC cr  nl
26CFS	KA1-RPTYP cr  nl
27C---	KA1-RPTNR cr  nl  cr  nl ^

T3440	cr  nl		
	cr  nl TN000010	TRANSACTION SEQUENCE NUMBER	cr  nl
	DATA	NAME	ERROR MESSAGE cr  n->
	01CHD	DEMSTAT-HEADER cr  nl	
	02C-----	ORGUIC cr  nl	
	03C---	RPTNR cr  nl	
	04C---	DATE cr  nl	
	05C----	TIME cr  nl	
	06C---	MONTH cr  nl	
	07C---	YEAR cr  nl	
	08C----	PIN cr  nl	
	09C-----	EXERC-OP-NAME cr  nl  cr  nl ^	

T3450	cr  nl		
	cr  nl TN000010	TRANSACTION SEQUENCE NUMBER	cr  nl
	DATA	NAME	ERROR MESSAGE cr  nl
	01CAS	SHORT-TONS cr  nl	
	02C---	TRANSNUM cr  nl	
	03C-----	PIN cr  nl	
	04C-----	ULN cr  nl	
	05C-----	UIC cr  nl	
	06C-----	STONBU cr  nl	
	07C-----	STONOV cr  nl	
	08C-----	STONOTS cr  nl	
	09C-----	STONNAT cr  nl  cr  nl ^	

T3460	cr  nl		
	cr  nl TN000010	TRANSACTION SEQUENCE NUMBER	cr  nl
	DATA	NAME	ERROR MESSAGE cr  nl
	01CAM	MEASUREMENT-TONS cr  nl	
	02C---	TRANSNUM cr  nl	
	03C-----	PIN cr  nl	
	04C-----	ULN cr  nl	
	05C-----	UIC cr  nl	
	06C-----	MTONBU cr  nl	
	07C-----	MTONOV cr  nl	
	08C-----	MTONOTS cr  nl	
	09C-----	MTONNAT cr  nl  cr  nl ^	

T3470	cr  nl
-------	--------



DATA	NAME	TRANSACTION SEQUENCE NUMBER	ERROR MESSAGE
U1[AV	VEHICLE-PAX		
U2[---	TRANSNUM		
U3[-----	PIN		
U4[-----	ULN		
U5[-----	UIC		
U6[-----	STONVEH		
U7[-----	MTONVEH		
U8[-----	SQFTUNVEH		
U9[-----	PAX		

T3480

DATA	NAME	TRANSACTION SEQUENCE NUMBER	ERROR MESSAGE
01[AE	VALIDATION		
02[---	TRANSNUM		
03[-----	PIN		
04[-----	ULN		
05[-----	UIC		
06[-	RTM		
07[-----	RTMEST		
08[---	RSNLA		

T3490

DATA	NAME	TRANSACTION SEQUENCE NUMBER	ERROR MESSAGE
U1[AT	TOA-PICKUP		
U2[---	TRANSNUM		
U3[-----	PIN		
U4[-----	ULN		
U5[-----	UIC		
U6[-----	RTMTOA		

T3500

DATA	NAME	TRANSACTION SEQUENCE NUMBER	ERROR MESSAGE
01[ED	EMPLOY-DEPART		
02[---	TRANSNUM		
03[-----	PIN		
04[-----	ULN		
05[-----	UIC		
06[-----	EMDD		

T3510

DATA	NAME	TRANSACTION SEQUENCE NUMBER	ERROR MESSAGE
U1[EA	EMPLOY-ARRIVE		
U2[---	TRANSNUM		
03[-----	PIN		

04C----- ULN|cr||nl|  
 05C----- UIC|cr||nl|  
 06C----- EMAD|cr||nl||cr||nl|^

T3520 |cr||nl|  
 |cr||nl|TNC000010 TRANSACTION SEQUENCE NUMBER |cr||nl|  
 DATA NAME ERROR MESSAGE|cr||nl|  
 01CMA ARRIVE-MOBSTATION|cr||nl|  
 U2C--- TRANSNUM|cr||nl|  
 U3C----- UIC|cr||nl|  
 U4C----- DTAMS|cr||nl|  
 U5C--- PPA|cr||nl||cr||nl|^

T3530 |cr||nl|  
 |cr||nl|TNC000010 TRANSACTION SEQUENCE NUMBER |cr||nl|  
 DATA NAME ERROR MESSAGE|cr||nl|  
 01CMC CHANGE-MOB-DATA|cr||nl|  
 U2C--- TRANSNUM|cr||nl|  
 U3C----- UIC|cr||nl|  
 U4C----- MDATE|cr||nl|  
 U5C----- MBODD|cr||nl|  
 U6C----- MBSAD|cr||nl|  
 U7C---- ACCEO|cr||nl|  
 U8C--- MBCMD|cr||nl||cr||nl|^

T3540 |cr||nl|  
 |cr||nl|TNC000010 TRANSACTION SEQUENCE NUMBER |cr||nl|  
 DATA NAME ERROR MESSAGE|cr||nl|  
 01CPL POE-LAST-DEPART|cr||nl|  
 U2C--- TRANSNUM|cr||nl|  
 U3C----- PIN|cr||nl|  
 U4C----- ULN|cr||nl|  
 U5C----- UIC|cr||nl|  
 U6C----- POELDEPART|cr||nl||cr||nl|^

T3550 |cr||nl|  
 |cr||nl|TNC000010 TRANSACTION SEQUENCE NUMBER |cr||nl|  
 DATA NAME ERROR MESSAGE|cr||nl|  
 01CPD POE-DEPART|cr||nl|  
 U2C--- TRANSNUM|cr||nl|  
 U3C----- PIN|cr||nl|  
 U4C----- ULN|cr||nl|  
 U5C----- UIC|cr||nl|  
 U6C----- POEDEPART|cr||nl|  
 U7C----- POECARRIER|cr||nl||cr||nl|^

T3560 |cr||nl|  
 |cr||nl|TNC000010 TRANSACTION SEQUENCE NUMBER |cr||nl|  
 DATA NAME ERROR MESSAGE|cr||nl|  
 01COL ORIGIN-LAST-DEPRT|cr||nl|

```

02[---          TRANSNUM|cr||nl|
03[-----      PIN|cr||nl|
04[-----      ULN|cr||nl|
05[-----      UIC|cr||nl|
06[-----      ORLDEPART|cr||nl||cr||nl|^

T3570  |cr||nl|
        |cr||nl|TNC000010      TRANSACTION SEQUENCE NUMBER |cr||nl|
        DATA      NAME      ERROR MESSAGE|cr||nl|
01[00          ORIGIN-DEPART|cr||nl|
02[---          TRANSNUM|cr||nl|
03[-----      PIN|cr||nl|
04[-----      ULN|cr||nl|
05[-----      UIC|cr||nl|
06[-----      ORDEPART|cr||nl|
07[-----      ORCARRIER|cr||nl||cr||nl|^

T6000  |nl||cr|SIOS ROUTINE|nl||cr|^

T6010  |nl||cr|^

T6020  |nl||cr|WHAT FUNCTION DO YOU WANT |nl||cr||nl||cr|
        MOVE, RETURN, CLEAR, SEARCH |nl||cr|^

T6030  m^

T6031  c|cr|^

T6032  cl^

T6033  s^

T6034  r^

T6035  |cr|^

T6100  |nl||cr|SIOS MOVE ROUTINE |nl||cr||nl||cr|
        YOUR FILE YOU WANT TO USE |nl||cr|^

T6110  ^

T6120  |nl||cr|SYSTEM ID YOU WANT TO USE |nl||cr|^

T6121  |nl||cr|YOUR FILE IS BUSY|nl||cr|^

T6122  |nl||cr|UNABLE TO ACCESS YOUR FILE|nl||cr|^

T6140  |nl||cr|WUFI FILE YOU WANT TO USE |nl||cr|^

T6141  |nl||cr|SYSTEM ID IS NOT AUTHORIZED |nl||cr|^

```

T6160 |nl||cr|DATA WAS MOVED|nl||cr|^  
 T6161 |nl||cr|SYSTEM ID IS INVALID ON WUFI|nl||cr|^  
 T6170 RECORDS|nl||cr|^  
 T6200 |nl||cr|SIOS RETURN ROUTINE |nl||cr||nl||cr|  
 YOUR FILE YOU WANT TO USE |nl||cr|^  
 T6260 |nl||cr|RECORDS WERE RETURNED |nl||cr|^  
 T6261 |nl||cr|NO DATA FOR THIS SYSTEM ID |nl||cr||nl||cr|  
 RECORDS WERE NOT RETURNED |nl||cr|^  
 T6280 |nl||cr|WUFI RECORDS WERE CLEARED |nl||cr|^  
 T6300 |nl||cr|SIOS CLEAR ROUTINE |nl||cr||nl||cr|  
 SYSTEM ID YOU WANT TO USE |nl||cr|^  
 T6340 |nl||cr|NO DATA TO CLEAR|nl||cr|^  
 T6400 |nl||cr|SIOS SEARCH ROUTINE |nl||cr||nl||cr|  
 SYSTEM ID YOU WANT TO USE |nl||cr|^  
 T6440 BLOCKS OF DATA |nl||cr|^  
 T6441 |nl||cr|NO RECORDS FOUND|nl||cr|^  
 T7000 ready|cr||nl|^  
 T7001 |cr||nl|illegal character^  
 T7002 |cr||nl|<52>CURRENT FILE NOT DEFINED|cr||nl|^  
 T9030 |cr||nl|\*^  
 T9040 run mc76/exercise/equip|cr|^  
 T9050 runs mc76/exercise/equip|cr|^  
 T9060 runj mc76/exercise/equip|cr|^  
 T9070 run mc76/exercise/rptmob|cr|^  
 T9075 runs mc76/exercise/rptmob|cr|^  
 T9080 runj mc76/exercise/rptmob|cr|^  
 T9090 run mc76/exercise/accimob|cr|^

T9100 runs mc76/exercise/accimob|cr|^  
T9110 runj mc76/exercise/accimob|cr|^  
T9120 run mc76/exercise/info|cr|^  
T9130 runs mc76/exercise/info|cr|^  
T9140 runj mc76/exercise/info|cr|^  
T9150 run mc76/exercise/strength|cr|^  
T9160 runs mc76/exercise/strength|cr|^  
T9170 runj mc76/exercise/strength|cr|^  
T9180 run mc76/exercise/ready|cr|^  
T9190 runs mc76/exercise/ready|cr|^  
T9200 runj mc76/exercise/ready|cr|^  
T9210 run mc76/exercise/pomcus|cr|^  
T9220 runs mc76/exercise/pomcus|cr|^  
T9230 runj mc76/exercise/pomcus|cr|^  
T9240 run mc76/exercise/deployrp|cr|^  
T9250 runs mc76/exercise/deployrp|cr|^  
T9260 runj mc76/exercise/deployrp|cr|^  
T9270 run mc76/exercise/dployhpa|cr|^  
T9280 runs mc76/exercise/dployhpa|cr|^  
T9290 runj mc76/exercise/dployhpa|cr|^  
T9300 run mc76/exercise/cinc|cr|^  
T9310 runs mc76/exercise/cinc|cr|^  
T9320 runj mc76/exercise/cinc|cr|^  
T9325 ^  
T9330 run mc76/exercise/exdeploy|cr|^

T9340 runs mc76/exercise/exdeploy|cr|^  
 T9350 runj mc76/exercise/exdeploy|cr|^  
 T9360 run mc76/exercise/trans|cr|^  
 T9370 runs mc76/exercise/trans|cr|^  
 T9380 runj mc76/exercise/trans|cr|^  
 T9390 run mc76/exercise/rptor|cr|^  
 T9400 runs mc76/exercise/rptor|cr|^  
 T9410 runj mc76/exercise/rptor|cr|^  
 T9420 run mc76/exercise/mbcoistl|cr|^  
 T9430 runs mc76/exercise/mbcoistl|cr|^  
 T9440 runj mc76/exercise/mbcoistl|cr|^  
 T9500 |cr||nl| .^  
 T9501 .|cr||nl|^  
 T9520 ENTER REQUIRED PARAMETER FOR 'OMNI-CATFILE'|cr||nl|=^  
 T9521 |cr||nl|ENTER REQUIRED PARAMETER FOR 'OMNI-CATFILE'|cr||nl|=^  
 T9530 |cr||nl|ENTER REQUIRED PARAMETER FOR 'WHERE-FLDNAME="LITERAL"'|cr||nl->|=^  
 T9531 |cr||nl|ENTER REQUIRED PARAMETER FOR 'WHERE-FIELDNAME="LITERAL"'|cr||->nl|=^  
 T9540 |cr||nl|ENTER REQUIRED PARAMETER FOR 'SORT-KEY1'|cr||nl|=^  
 T9550 |cr||nl|ENTER REQUIRED PARAMETER FOR 'SORT-KEY2'|cr||nl|=^  
 T9560 |cr||nl|ENTER REQUIRED PARAMETER FOR 'OUTPUT-CATFILE'|cr||nl|=^  
 T9570 ENTER REQUIRED PARAMETER FOR '"SBRPT"'|cr||nl|=^  
 T9580 |cr||nl|ENTER REQUIRED PARAMETER FOR 'SORT-FLD1-OR-BLANKS'|cr||nl|=^  
 T9590 |cr||nl|ENTER REQUIRED PARAMETER FOR 'SORT-FLD2-OR-BLANKS'|cr||nl|=^  
 T9600 |cr||nl|ENTER REQUIRED PARAMETER FOR 'SORT-FLD3-OR-BLANKS'|cr||nl|=^

T9700 RUN-ID? ^  
 T9710 |cr||nl|\$ IDENT? ^  
 T9750 |cr||nl|MUST BE 12 CHARACTERS OR LESS^  
 T9754 |cr||nl|SNUMB NOT IN SYSTEM^  
 T9755 |cr||nl|LOOK, MORE, DELE, ABRT, QUIT? ^  
 T9756 |cr||nl|SNUMB(S)- ^  
 T9757 |cr||nl|SNUMB? ^  
 T9758 |cr||nl|SNUMB NOT IN TABLE|cr||nl|^  
 T9759 |cr||nl|snumb?^  
 T9760 look|cr|^  
 T9761 more|cr|^  
 T9762 dele^  
 T9763 abrt|cr|^  
 T9764 quit|cr|^  
 T9780 |cr||nl|please direct, release, or hold before exit|cr||nl|^  
 T9800 JOUT INVOKED FOR SNUMB^  
 T9801 |cr||nl|function?^  
 T9802 print \$\$|cr|^  
 T9803 prin \$\$|cr|^  
 T10170 end of \$\$^  
 T10180 hold|cr|^  
 T10190 rele^  
 T10200 direct ac|cr|^  
 T10205 direct onl|cr|^  
 T10210 normal termination ^

T10230 print 74|cr|^  
 T10240 prin 74|cr|^  
 T10250 end of 74^  
 T10260 |cr||nl|74 not found^  
 T10270 output not found^  
 T10280 output busy^  
 T10290 NOT IN SYSTEM^  
 T10300 INVOKE JOUT MANUALLY^  
 T11000 |cr||nl||ff||cr||nl|TELECONFERENCING AT ^  
 T11001 |cr||nl|ILLEGAL CLASSIFICATION CODE^  
 T11010 |cr||nl|USER ASSISTANCE IS AVAILABLE IN TALK AND COMMAND MODES. |cr||->  
 nl|  
 THE QUESTION MARK (?) WILL INTRODUCE YOU TO THIS FEATURE. |cr||nl|  
 |cr||nl|INITIATE, RECONVENE, OR JOIN?^  
 T11020 j|cr|^  
 T11021 jo|cr|^  
 T11022 joi|cr|^  
 T11023 join|cr|^  
 T11030 |cr||nl|NAME OF CONFERENCE YOU WISH TO JOIN?^  
 T11031 LINE TERMINATED^  
 T11040 ^  
 T11050 |cr||nl|PARTICIPANT'S NAME? ^  
 T11051 |cr||nl|SORRY, THAT CONFERENCE IS NOT IN SESSION^  
 T11070 |cr||nl|PARTICIPANT'S PASSWORD? |cr||nl|  
 \*%&<%#>?\*#\$%#?AM\*5\*48#N&#|cr|  
 TXOGWBMWEPVRMCBQNKMGAKZMA|cr|^  
 T11071 |cr||nl|SORRY, NAME GIVEN NOT RECOGNIZED^  
 T11072 |cr||nl|ACCESS DENIED ^



T11090 |cr||nl|THE FOLLOWING QUESTIONS CONCERN TRANSCRIPT DELIVERY |cr||nl|  
 ENTER HOST NAME FOR DELIVERY OF YOUR COPY ^  
 T11110 |cr||nl|ENTER YOUR ACCOUNT NUMBER, NAME FOR |cr||nl|  
 IDENT CARD ^  
 T11091 |cr||nl|ACCESS GRANTED ^  
 T11111 |cr||nl|ENTER YOUR ACCOUNT NUMBER, NAME FOR |cr||nl|  
 IDENT CARD—^  
 T11120 .^  
 T11140 |cr||nl|CONFERENCE SECURITY CLASSIFICATION CODE IS ^  
 T11150 |nl|THERE ARE NO ANNOUNCEMENTS AT THIS TIME |cr|^  
 T11151 |nl|ANNOUNCEMENT MESSAGE NUMBERS ARE|cr||nl|^  
 T11160 |nl|LATEST MESSAGE NUMBER IS^  
 T11170 |nl|LAST MESSAGE SEEN WAS NUMBER^  
 T11180 |nl||ff||cr||nl|LISTEN MODE ENTERED IN ^  
 T11190 HAS JOINED THE CONFERENCE ^  
 T11191 HAS LEFT THE CONFERENCE ^  
 T11500 |cr||nl||ff||cr||nl|LISTEN MODE ENTERED IN ^  
 T11510 |ff|^  
 T11600 |cr||nl||ff|^  
 T12000 |cr||nl|COMMAND ?^  
 T12001 |cr||nl||ff||cr||nl|COMMAND ?^  
 T12010 bull^  
 T12011 comm^  
 T12012 defi^  
 T12013 floo^  
 T12014 give^  
 T12015 list^

T12016 prin^  
T12017 quit^  
T12018 revi^  
T12019 stat^  
T12020 talk^  
T12021 ?^  
T12022 desc^  
T12040 |cr||nl|NO MESSAGES WAITING^  
T12041 |cr||nl||ff||cr||nl|MESSAGE NUMBER^  
T12050 COMMAND ?^  
T12060 |cr||nl|ENTER ONE LINE PER PROMPT (-), 13 LINES MAX |cr||nl|  
NULL INPUT OR ALL SPACES = END OF MESSAGE ^  
T12061 |cr||nl|INVALID COMMUNICATION ARGUMENT ^  
T12070 |cr||nl|-^  
T12071 |cr||nl|COMMUNICATION SENT ^  
T12072 |cr||nl|COMMUNICATION REQUEST IGNORED ^  
T12080 |cr|^  
T12100 |cr||nl|FLOOR RECORDING IS OFF |cr||nl|  
COMMAND ?^  
T12101 |cr||nl|FLOOR RECORDING IS ON |cr||nl|  
COMMAND ?^  
T12102 |cr||nl|CONFERENCE IS OPEN TO JOINING PARTICIPANTS |cr||nl|  
COMMAND ?^  
T12103 |cr||nl|CONFERENCE IS CLOSED TO JOINING PARTICIPANTS |cr||nl|  
COMMAND ?^  
T12104 |cr||nl|OPTION RESTRICTED TO INITIATING CHAIRMAN|cr||nl|  
COMMAND ?^  
T12105 |cr||nl|OPTION RESTRICTED TO CURRENT CHAIRMAN |cr||nl|  
COMMAND ?^

T12106 |cr||nl|NO ALTERNATES HAVE BEEN ASSIGNED|cr||nl|  
 COMMAND ?^  
 T12107 CHANGE? |cr||nl|^  
 T12108 FILE INSERT LIMIT IS-- ^  
 T12120 |cr||nl|HOST? |cr||nl|^  
 T12121 |cr||nl|\$IDENT? |cr||nl|^  
 T12122 |cr||nl|NEW IDENT CARD NOT VALID AT THIS HOST |cr||nl|  
 CHANGE? |cr||nl|^  
 T12200 |cr||nl|ENTER ONE LINE OF TEXT TO BE SENT WITH YOUR REQUEST^  
 T12201 |cr||nl|SORRY, CHAIRMAN NOT PRESENT TO GRANT FLOOR PRIVILEGE^  
 T12202 |cr||nl|FLOOR INVALID COMMAND^  
 T12203 |cr||nl|INVALID FLOOR COMMAND ARGUMENT--^  
 T12210 |cr||nl|(ENTER BREAK STATUS OR NULL RESPONSE IF TEXT IS NOT DESIRED)^  
 T12230 |cr||nl|CHAIRMAN'S APPROVAL IS BEING REQUESTED, CONTINUE YOUR ACTIVIT->  
 Y^  
 T12240 |cr||nl|COMMAND VALID FOR FLOOR ONLY^  
 T12250 |cr||nl|PRINT REQUEST INCOMPLETE OR INVALID ^  
 T12251 |cr||nl|MESSAGE NUMBER TOO HIGH ^  
 T12252 CHANGE?^  
 T12270 |cr||nl|ENTER HOST NAME FOR DELIVERY OF YOUR COPY ^  
 T12271 |cr||nl|INVALID HOST NAME ^  
 T12272 |cr||nl|CHANGE? ^  
 T12273 |cr||nl|ENTER YOUR ACCOUNT NUMBER,NAME FOR IDENT CARD |cr||nl|^  
 T12274 |cr||nl|IDENT INFORMATION NOT VALID AT THIS HOST|cr||nl|  
 DO YOU WISH TO CHANGE IT(YES OR NO)?^  
 T12350 |cr||nl|QUIT SUCCESSFUL ^  
 T12360 |cr||nl|INVALID REVIEW REQUEST^

T12470 |cr||nl||cr||nl|THIS TERMINAL IS CURRENTLY IN COMMAND MODE.|cr||nl||c->  
 r||nl|  
 COMMAND MODE ALLOWS A PARTICIPANT TO ENTER OTHER MODES, TO|cr||nl|  
 OBTAIN CONFERENCE INFORMATION AND TO PERFORM SPECIFIC FUNCTIONS.|cr||->  
 nl|  
 A SPECIFIC COMMAND CAPABILITY IS ACQUIRED BY RESPONDING WITH A|cr||nl->  
 |  
 COMMAND VERB TO THE PROMPT 'COMMAND?'. THE POSSBLE RESPONSES|cr||nl|  
 ARE:|cr||nl||cr||nl|  
 ADD ADJOURN ANNOUNCE BULLETIN CHAIR TO|->  
 cr||nl|  
 COMMUNICATION DEFINITION DELETE DESCRIBE ERASE|cr||->  
 |nl|  
 FLOOR FORMAT GIVEUP INSERT KEYWORD|c->  
 r||nl|  
 LISTEN PDAC PRINT QUIT REVIEW|cr->  
 ||nl|  
 STATUS TALK TERMINATE ?|cr||nl||cr||nl|  
 AN EXPLANATION OF EACH VERB IS POSSIBLE BY RESPONDING|cr||nl|  
 'DESCRIBE VERB'--WHERE VERB IS ANY VALID COMMAND MODE VERB.|cr||nl||c->  
 r||nl|

EXAMPLE|cr||nl|  
 COMMAND?DESCRIBE FLOOR|cr||nl|  
 |cr||nl|COMMAND ?^

T12500 |cr||nl||ff||cr||nl|>>>ADD<<< |cr||nl|^  
 T12501 |cr||nl||ff||cr||nl|>>>ADJOURN<<< |cr||nl|^  
 T12502 |cr||nl||ff||cr||nl|>>>ANNOUNCE<<< |cr||nl|^  
 T12503 |cr||nl||ff||cr||nl|>>>BULLETIN<<< |cr||nl|^  
 T12504 |cr||nl||ff||cr||nl|>>>CHAIR TO<<< |cr||nl|^  
 T12505 |cr||nl||ff||cr||nl|>>>COMMUNICATION<<< |cr||nl|^  
 T12506 |cr||nl||ff||cr||nl|>>>DEFINITION<<<|cr||nl|^  
 T12507 |cr||nl||ff||cr||nl|>>>DELETE<<<|cr||nl|^  
 T12508 |cr||nl||ff||cr||nl|>>>DESCRIBE<<< |cr||nl|^  
 T12509 |cr||nl||ff||cr||nl|>>>ERASE<<< |cr||nl|^  
 T12510 |cr||nl||ff||cr||nl|>>>FLOOR<<< |cr||nl|^  
 T12511 |cr||nl||ff||cr||nl|>>>FORMAT<<<|cr||nl|^  
 T12512 |cr||nl||ff||cr||nl|>>>GIVEUP<<< |cr||nl|^

T12513 |cr||nl||ff||cr||nl|>>>INSERT<<< |cr||nl|^

T12514 |cr||nl||ff||cr||nl|>>>KEYWORD<<< |cr||nl|^

T12515 |cr||nl||ff||cr||nl|>>>LISTEN<<< |cr||nl|^

T12516 |cr||nl||ff||cr||nl|>>>PDAC<<< |cr||nl|^

T12517 |cr||nl||ff||cr||nl|>>>PRINT<<< |cr||nl|^

T12518 |cr||nl||ff||cr||nl|>>>QUIT<<< |cr||nl|^

T12519 |cr||nl||ff||cr||nl|>>>REVIEW<<< |cr||nl|^

T12520 |cr||nl||ff||cr||nl|>>>STATUS<<< |cr||nl|^

T12521 |cr||nl||ff||cr||nl|>>>TALK<<< |cr||nl|^

T12522 |cr||nl||ff||cr||nl|>>>TERMINATE<<< |cr||nl|^

T12523 |cr||nl||ff||cr||nl|>>> ? <<< |cr||nl|^

T12524 |cr||nl|INVALID DESCRIBE ARGUMENT PASSED^

T12600 |cr||nl|THIS VERB'S USAGE IS RESTRICTED TO THE CURRENT CHAIRMAN AND A->  
 LLOWS|cr||nl|  
 HIM TO ADD TO THE VALID PARTICIPANT LIST FOR THE CONFERENCE. IT|cr||->  
 nl|  
 ALSO ALLOWS HIM TO CORRECT/CHANGE THE USERID ASSOCIATED WITH ..|cr||nl->  
 |  
 PARTICULAR PARTICIPANT NAME. A PARTICIPANT NAME CANNOT BE CHANGED.|c->  
 r||nl|  
 THE USERID AND PARTICIPANT NAME TO BE ADDED/CORRECTED MUST FOLLOW|cr||->  
 nl|  
 THE VERB IN THE SAME FORMAT USED DURING CONFERENCE INITIATION|cr||nl|  
 DEFINITION.|cr||nl||cr||nl|  
 EXAMPLE|cr||nl|  
 COMMAND?ADD USERID/PNAME|cr||nl|  
 |cr||nl|COMMAND ?^

T12610 |cr||nl|THIS VERB'S USAGE IS RESTRICTED TO THE CURRENT CHAIRMAN AND A->  
 LLOWS|cr||nl|  
 HIM TO ADJOURN A CONFERENCE SESSION. AN ADJOURNED CONFERENCE CAN|cr||->  
 nl|  
 BE RECONVENED AT A LATER TIME. AN ADJOURNMENT WILL NOT TAKE PLACE|cr->  
 ||nl|  
 IF THERE ARE ACTIVE PARTICIPANTS OTHER THAN THE CHAIRMAN STILL|cr||nl->  
 |  
 CONNECTED TO THE CONFERENCE. NO PARTICIPANTS ARE PERMITTED TO JOIN|c->  
 r||nl|

THE CURRENT SESSION OF THE CONFERENCE ONCE THE ADJOURN IS ENTERED.|cr->  
||nl|

EXAMPLE|cr||nl|  
COMMAND?ADJOURN|cr||nl|  
|cr||nl|COMMAND ?"

T12620 |cr||nl|THIS VERB'S USAGE IS RESTRICTED TO THE CURRENT CHAIRMAN AND A->  
LLOWS|cr||nl|  
HIM TO CREATE A LIST OF MESSAGES TO BE DISPLAYED TO PARTICIPANTS|cr||->  
nl|  
UPON JOINING. IT ALLOWS HIM TO CHANGE OR DELETE THE LIST.|cr||nl|  
ARGUMENTS FOR THE VERB ARE A LIST OF UP TO 10 VALID MESSAGE NUMBERS|c->  
r||nl|  
DELIMITED BY SEMICOLONS. IF NO ARGUMENTS ARE FOUND THE ANNOUNCEMENT|->  
cr||nl|  
LIST IS PURGED.|cr||nl||cr||nl|  
EXAMPLE 1|cr||nl|  
COMMAND?ANNOUNCE 2;3;5|cr||nl||cr||nl|  
(MESSAGES 2, 3, 5 WILL BE ANNOUNCED TO NEW PARTICIPANTS AS|cr||nl|  
THEY JOIN.)|cr||nl||cr||nl|  
EXAMPLE 2|cr||nl|  
COMMAND?ANNOUNCE|cr||nl||cr||nl|  
(ANY PREVIOUS ANNOUNCEMENT LIST WILL BE ERASED.)|cr||nl|  
|cr||nl|COMMAND ?"

T12630 |cr||nl|THIS VERB ALLOWS AN INDIVIDUAL PARTICIPANT TO SEE THE NEXT ME->  
SSAGE|cr||nl|  
QUEUED TO HIS TERMINAL WITHOUT ENTERING LISTEN MODE. ONLY ONE|cr||nl->  
|  
MESSAGE WILL BE RECEIVED PER EACH BULLETIN REQUEST.|cr||nl||cr||nl|  
EXAMPLE|cr||nl|  
COMMAND?BULLETIN|cr||nl||cr||nl|  
|cr||nl|COMMAND ?"

T12640 |cr||nl|THIS VERB'S USAGE IS RESTRICTED TO THE CHAIRMAN (CURRENT OR|c->  
r||nl|  
ORIGINAL). THE CURRENT CHAIRMAN MAY PASS THE CHAIRMANSHIP TO|cr||nl|  
ANOTHER ACTIVE PARTICIPANT. THE ORIGINAL CHAIRMAN CAN AT ANY TIME|cr->  
||nl|  
RETAKE THE CHAIRMANSHIP HE PREVIOUSLY PASSED.|cr||nl||cr||nl|  
EXAMPLE 1|cr||nl|  
COMMAND?CHAIR TO PARTICIPANT NAME|cr||nl||cr||nl|  
(THE CHAIRMANSHIP WILL BE GIVEN TO THE NAMED PARTICIPANT.)|cr||nl||cr->  
||nl||cr||nl|  
EXAMPLE 2|cr||nl|  
COMMAND?CHAIR TO|cr||nl||cr||nl|  
(FORM OF VERB INITIATING CHAIRMAN WOULD USE TO RETAKE THE|cr||nl|  
CHAIRMANSHIP.)|cr||nl||cr||nl|  
ANOTHER FUNCTION OF THIS COMMAND IS TO ALLOW A DESIGNATED ALTERNATE|c->  
r||nl|

TO ASSUME THE CHAIRMANSHIP. THIS MAY ONLY BE DONE IF THE CONFERENCE|->  
cr||nl|  
IS WITHOUT A CHAIRMAN AT THE TIME THE COMMAND IS INVOKED.|cr||nl||cr|->  
nl|

EXAMPLE 3|cr||nl|  
COMMAND?CHAIR TO|cr||nl||cr||nl|  
(THE DESIGNATED ALTERNATE WILL RECEIVE THE CHAIR IF IT IS VACANT.)  
|cr||nl|COMMAND ?^

T12650 |cr||nl|THIS VERB WILL ALLOW INFORMAL MESSAGE TEXT TO BE SENT TO SEVE->  
RAL|cr||nl|  
OR ALL ACTIVE PARTICIPANTS. IT DIFFERS FROM A TALK MESSAGE IN THAT|c->  
r||nl|  
THE MESSAGE IS NOT WRITTEN TO THE TRANSCRIPT AND IS DELIVERED WITH|cr->  
||nl|  
PRIORITY MESSAGES. THIS MEANS THE RECIPIENT DOES NOT HAVE TO BE IN|c->  
r||nl|  
LISTEN MODE TO RECEIVE THE COMMUNICATION.|cr||nl||cr||nl|

EXAMPLE|cr||nl|  
COMMAND?COMMUNICATION \$ALL|cr||nl|  
ENTER TEXT WHEN PROMPT CHARACTER APPEARS (-),|cr||nl|  
MAXIMUM INPUT IS 13 LINES|cr||nl|  
NULL INPUT OR ALL SPACES INDICATES END OF MESSAGE|cr||nl|  
-THIS IS AN EXAMPLE OF A GENERAL COMMUNICATION REQUEST|cr||->  
nl|

-(CR)|cr||nl||cr||nl|  
EXAMPLE|cr||nl|  
COMMAND?COMMUNICATION TO PARTICIPANT1|cr||nl|  
ENTER TEXT WHEN PROMPT CHARACTER APPEARS (-),|cr||nl|  
MAXIMUM INPUT IS 13 LINES|cr||nl|  
NULL INPUT OR ALL SPACES INDICATES END OF MESSAGE|cr||nl|  
-THIS MESSAGE WILL BE DELIVERED TO PARTICIPANT 1|cr||nl|  
-(CR)|cr||nl|  
|cr||nl|COMMAND ?^

T12660 |cr||nl|THIS VERB DISPLAYS THE STATE OF SEVERAL CONFERENCE PARAMETERS->  
. IN|cr||nl|  
SOME CASES IT ALLOWS THE STATE TO BE CHANGED. A REQUEST MAY BE|cr||n->  
l|  
GENERAL OR SPECIFIC. A GENERAL REQUEST WILL DISPLAY ALL INFORMATION|->  
cr||nl|  
TO WHICH THE USER IS ENTITLED. A SPECIFIC REQUEST WILL DISPLAY THE|c->  
r||nl|  
REQUESTED INFORMATION, AND PROMPT FOR CHANGE WHEN CHANGE IS POSSI-|cr->  
||nl|  
BLE.|cr||nl||cr||nl|  
DEFINITION -- THOSE CONFERENCE PARAMETERS AVAILABLE FOR|cr|->  
nl|

VIEWING BY THE INDIVIDUAL ARE DISPLAYED|cr||nl|  
DEFINITION SCC -- DISPLAYS CONFERENCE SECURITY CLASSIFICA|->

cr||nl| TION; CAN BE CHANGED BY THE CURRENT CHAIRMAN|cr||nl|  
 DEFINITION ACCESS -- INFORMS AS TO WHETHER NEW PARTICIPANTS->  
 MAY JOIN;|cr||nl|  
 CAN BE CHANGED BY THE CURRENT CHAIRMAN|cr||nl|  
 DEFINITION FLOOR -- INFORMS AS TO WHETHER FLOOR RECORDING I->  
 S TAKING|cr||nl|  
 PLACE; CAN BE CHANGED BY THE CURRENT CHAIRMAN|cr||nl|  
 DEFINITION FILE -- DISPLAYS THE NAME OF THE TRANSCRIPT FILE->  
 |cr||nl|  
 DEFINITION ANNOUNCE -- DISPLAYS THE ANNOUNCEMENT MESSAGE|cr->  
 ||nl|  
 NUMBERS; CAN BE CHANGED BY THE CURRENT CHAIRMAN USING|->  
 cr||nl|  
 COMMAND MODE VERB ANNOUNCE|cr||nl|  
 DEFINITION INSERT -- DISPLAYS CURRENT INSERT FILE LINE|cr||->  
 nl|  
 LIMIT; CAN BE CHANGED BY CURRENT CHAIRMAN USING|cr||nl->  
 |  
 COMMAND MODE VERB INSERT|cr||nl|  
 DEFINITION KEYWORD -- TELLS HOW MANY KEYWORDS ARE CURRENT-|->  
 cr||nl|  
 LY ASSIGNED; CAN BE CHANGED BY THE CURRENT CHAIRMAN|cr->  
 ||nl|  
 USING COMMAND MODE VERB KEYWORD|cr||nl|  
 DEFINITION CHAIR -- DISPLAYS TO THE INITIATING CHAIRMAN|cr|->  
 |nl|  
 HIS PASSWORD AND PROMPTS HIM FOR CHANGE|cr||nl|  
 DEFINITION PARTICIPANT -- DISPLAYS TO THE CHAIRMAN THE|cr||->  
 nl|  
 CURRENT PARTICIPANT PASSWORD AND PROMPTS FOR CHANGE|cr->  
 ||nl|  
 DEFINITION LIST -- DISPLAYS A LIST OF ALL VALID PARTICI-|cr->  
 ||nl|  
 PANTS, FOR THE CURRENT CHAIRMAN IT ALSO DISPLAYS THE|c->  
 r||nl|  
 ASSOCIATED USERID; CAN BE CHANGED BY THE CURRENT|cr||n->  
 l|  
 CHAIRMAN USING COMMAND MODE VERB ADD|cr||nl|  
 DEFINITION DELIVERY -- DISPLAYS USER TRANSCRIPT DELIVERY|cr->  
 ||nl|  
 INFORMATION PROVIDED DURING JOIN PROCESSING AND|cr||nl->  
 |  
 PROMPTS FOR CHANGE|cr||nl|  
 DEFINITION ALTERNATE -- DISPLAYS THE NAMES OF THOSE PARTI-|->  
 cr||nl|  
 CIPANTS WHO HAVE BEEN DESIGNATED AS ALTERNATE CHAIRMAN->  
 ;|cr||nl|  
 CAN BE CHANGED BY THE CURRENT CHAIRMAN|cr||nl||cr||nl|  
 EXAMPLE|cr||nl|



COMMAND?DEFINITION DELIVERY|cr||nl|  
 DELIVERY INSTRUCTIONS FOR YOUR COPY OF TRANSCRIPT|cr||nl|  
 HOST-HOST NAME|cr||nl|  
 \$IDENT-\$IDENT INFORMATION|cr||nl|  
 CHANGE?YES|cr||nl|  
 HOST?HOST NAME|cr||nl|  
 \$IDENT-(NULL, WILL NOT CHANGE)|cr||nl||cr||nl|  
 |cr||nl|COMMAND ?"

T12670 |cr||nl|THIS COMMAND ALLOWS THE CHAIRMAN TO REMOVE AN UNUSED PARTICIP-  
 ANT|cr||nl|  
 NAME FROM THE LIST OF NAMES ACCEPTABLE TO THE CONFERENCE. A LIST OF|->  
 cr||nl|  
 UNUSED NAMES IS AVAILABLE BY INVOKING THE COMMAND WITHOUT AN|cr||nl|  
 ARGUMENT.|cr||nl||cr||nl|  
 EXAMPLE 1|cr||nl|  
 COMMAND?DELETE|cr||nl|  
 (LIST OF UNUSED PARTICIPANT NAMES)|cr||nl||cr||nl|  
 EXAMPLE 2|cr||nl|  
 COMMAND?DELETE PARTICIPANT5|cr||nl|  
 |cr||nl|COMMAND ?"

T12680 |cr||nl|THIS VERB IS USED TO OBTAIN AN EXPLANATION OF COMMAND MODE|cr->  
 ||nl|  
 VERBS. TO RECEIVE AN EXPLANATION OF A VERB, THE|cr||nl|  
 VERB WOULD FOLLOW THE WORD 'DESCRIBE'. THE VERBS THAT CAN BE|cr||nl|  
 DESCRIBED ARE:|cr||nl||cr||nl|  
 ADD                      ADJOURN                      ANNOUNCE                      BULLETIN                      CHAIR TO|->  
 cr||nl|  
 COMMUNICATION      DEFINITION                      DELETE                      DESCRIBE                      ERASE|cr|->  
 |nl|  
 FLOOR                      FORMAT                      GIVEUP                      INSERT                      KEYWORD|c->  
 r||nl|  
 LISTEN                      PDAC                      PRINT                      QUIT                      REVIEW|cr->  
 ||nl|  
 STATUS                      TALK                      TERMINATE                      ?|cr||nl||cr||nl|  
 EXAMPLE|cr||nl|  
 COMMAND?DESCRIBE FLOOR|cr||nl|  
 |cr||nl|COMMAND ?"

T12690 |cr||nl|THIS COMMAND IS USED BY THE CHAIRMAN TO REMOVE THE TEXT OF A ->  
 MESSAGE|cr||nl|  
 FROM THE CONFERENCE TRANSCRIPT. THE MESSAGE BANNER WILL BE RETAINED|->  
 cr||nl|  
 WITH A NOTE STATING THAT THE MESSAGE HAS BEEN DELETED. A MESSAGE|cr|->  
 |nl|  
 BEING READ CANNOT BE ERASED.|cr||nl||cr||nl|  
 EXAMPLE|cr||nl|  
 COMMAND?ERASE 5|cr||nl|  
 |cr||nl|COMMAND ?"

T12700 |cr||nl|THE PURPOSE OF THE FLOOR IS TO ALLOW A PARTICIPANT TO ADDRESS->  
 |cr||nl|  
 MULTIPLE PARTICIPANTS AT THE SAME TIME. THE FLOOR PRIVILEGE CAN|cr||->  
 nl|  
 ONLY BE ATTAINED BY DIRECT ACTION ON THE PART OF THE CHAIRMAN. THIS|->  
 cr||nl|  
 ACTION CAN BE AT THE CHAIRMAN'S INITIATIVE OR PROMPTED BY A|cr||nl|  
 PARTICIPANT REQUEST.|cr||nl||cr||nl|  
 FLOOR -- REQUEST BY PARTICIPANT OR CHAIRMAN FOR THE FLOOR.|->  
 cr||nl|  
 THE PARTICIPANT MUST WAIT FOR INTERVENTION BY THE|cr||->  
 nl|  
 CHAIRMAN. THE CHAIRMAN WOULD AUTOMATICALLY RECEIVE|cr->  
 ||nl|  
 THE FLOOR|cr||nl|  
 FLOOR [TO] PARTICIPANT NAME -- WITH THIS FORM OF THE|cr||nl->  
 |  
 COMMAND THE CHAIRMAN CAN GIVE THE FLOOR TO A PARTICI|->  
 cr||nl|  
 PANT. IF THE PARTICIPANT ACCEPTS RECEIPT OF THE|cr||n->  
 l|  
 FLOOR, ANYONE CURRENTLY OCCUPYING THE FLOOR WOULD|cr||->  
 nl|  
 IT|cr||nl|  
 FLOOR FREE -- WITH THIS FORM OF THE COMMAND THE CHAIRMAN|cr->  
 ||nl|  
 CAN TAKE THE FLOOR FROM WHOMEVER HAS IT AND LEAVE IT|c->  
 r||nl|  
 VACANT|cr||nl|  
 FLOOR CANCEL -- A PARTICIPANT WOULD USE THIS FORM OF THE|cr->  
 ||nl|  
 COMMAND TO CANCEL A FLOOR REQUEST NOT YET ACTED UPON|c->  
 r||nl|  
 BY THE CHAIRMAN|cr||nl||cr||nl|  
 EXAMPLE|cr||nl|  
 COMMAND?FLOOR TO PARTICIPANT NAME|cr||nl||cr||nl|  
 (FLOOR WILL BE TAKEN FROM WHOMEVER MIGHT HAVE IT, AND GIVEN TO THE|cr->  
 ||nl|  
 NAMED PARTICIPANT IF HE ACCEPTS THE FLOOR)|cr||nl|  
 |cr||nl|COMMAND ?^

T12710 |cr||nl|THIS COMMAND IS USED BY THE CHAIRMAN TO ESTABLISH THE HEADER ->  
 Tu|cr||nl|  
 APPEAR AT THE BEGINNING OF TALK MESSAGES. THE COMMAND WILL PROMPT|cr->  
 ||nl|  
 FOR INPUT WHEN THE OPTIONS DESIRED ARE NOT PART OF THE INITIAL|cr||nl->  
 |  
 REQUEST. THE OPTIONS ARE IMPLEMENTED IN THE ORDER RECEIVED. END OF|->  
 cr||nl|  
 REQUEST IS INDICATED BY TRANSMITTING SEND OR EDM TO THE INPUT PROMPT.->

|cr||nl|  
 THE OPTIONS MAY BE CHANGED AT ANYTIME. THEY MAY BE CANCELLED BY USE|->  
 cr||nl|  
 OF THE ARGUMENT CANCEL. THE OPTIONS ARE:|cr||nl||cr||nl|  
 CLASSIFICATION DATE-TIME-GROUP FROM INFORMATION|cr||nl->  
 |  
 REFERENCE SUBJECT TO|cr||nl||cr||nl|  
 EXAMPLE 1|cr||nl|  
 COMMAND?FORMAT TO;FROM;SUBJECT;\$END|cr||nl||cr||nl|  
 EXAMPLE 2|cr||nl|  
 COMMAND?FORMAT CANCEL|cr||nl|  
 |cr||nl|COMMAND ?^

T12720 |cr||nl|THIS VERB ALLOWS A PARTICIPANT TO RELINQUISH THE FLOOR|cr||nl->  
 |  
 PRIVILEGE. IT IS INVALID IF THE PARTICIPANT DOES NOT HAVE THE|cr||nl->  
 |  
 FLOOR.|cr||nl||cr||nl|  
 EXAMPLE|cr||nl||cr||nl|  
 COMMAND?GIVEUP|cr||nl|  
 |cr||nl|COMMAND ?^

T12730 |cr||nl|THIS IS A CHAIRMAN ONLY VERB. IT ALLOWS HIM TO CONTROL THE N->  
 UMBER|cr||nl|  
 OF LINES THAT CAN BE INSERTED INTO A MESSAGE FROM AN EXTERNAL FILE.|c->  
 r||nl|  
 THE LINES CAN BE RESTRICTED TO A VALUE BETWEEN 1 AND 600, OR ANY|cr||->  
 nl|  
 RESTRICTION CAN BE COMPLETELY ELIMINATED.|cr||nl||cr||nl|  
 EXAMPLE 1|cr||nl|  
 COMMAND?INSERT 10|cr||nl||cr||nl|  
 EXAMPLE 2|cr||nl|  
 COMMAND?INSERT ELIMINATE|cr||nl||cr||nl|  
 NOTE:NO MORE THAN 600 FULL 80 CHARACTER LINES MAY MAKE UP A CON-|cr||->  
 nl|  
 FERENCE MESSAGE. IF THE AVERAGE LINE CHARACTER COUNT IS LESS THAN|cr->  
 ||nl|  
 80, THE MAXIMUM POSSIBLE LINES WOULD INCREASE. IT IS ONLY IN THESE|c->  
 r||nl|  
 CASES THAT ELIMINATING LINE COUNT HAS ANY MEANING.|cr||nl|  
 |cr||nl|COMMAND ?^

T12740 |cr||nl|MESSAGES MAY BE ASSIGNED A SUBJECT LINE WHEN THEY ARE BUILT. ->  
 THE|cr||nl|  
 SUBJECT LINE PROVIDES FOR TOPIC ORIENTATION AND SELECTIVE REVIEWING.|->  
 cr||nl|  
 TO CONTROL THE CONFERENCE SUBJECTS THE CHAIRMAN WOULD USE THE|cr||nl|  
 VERB 'KEYWORD' TO BUILD A VALID SUBJECT LIST. ONLY MESSAGES WHOSE|cr->  
 ||nl|  
 SUBJECT LINE CONTAINS A WORD FROM THIS LIST WILL BE REVIEWABLE BY|cr|->

```

|nl|
SUBJECT.|cr||nl||cr||nl|
      EXAMPLE|cr||nl|
      COMMAND?KEYWORD|cr||nl|
      ITEM?TANKS|cr||nl|
      THE SUBJECT WORD TO BE ADDED IS-TANKS|cr||nl|
      (YES OR NO)?YES|cr||nl|
      SUBJECT WORD ACCEPTED|cr||nl|
|cr||nl|COMMAND ?~

```

T12750 |cr||nl|TO RECEIVE CONFERENCE MESSAGES AND VIEW FLOOR ACTIVITY A PART->  
ICIPANT|cr||nl|  
MUST BE IN LISTEN MODE. THIS VERB TRANSFERS A PARTICIPANT FROM|cr||n->  
L|  
COMMAND MODE TO LISTEN MODE. IT IS INVALID IF THE PARTICIPANT HAS|cr->  
||nl|  
THE FLOOR.|cr||nl||cr||nl|  
 EXAMPLE|cr||nl|  
 COMMAND?LISTEN|cr||nl||cr||nl|  
NOTE: IT IS NOT HONORED IF THE PARTICIPANT HAS THE FLOOR.|cr||nl|  
|cr||nl|COMMAND ?~

T12760 |cr||nl|THIS VERB OPENS A CONNECTION TO A DIRECT ACCESS PROGRAM IN TH->  
E SAME|cr||nl|  
HOST. THE VERB MUST BE FOLLOWED BY THE NAME OF THE DIRECT ACCESS|cr|->  
|nl|  
PROGRAM TO BE CONNECTED TO. WHEN NECESSARY A PDAC CONNECTION CAN BE|->  
cr||nl|  
SUSPENDED. THIS IS ACCOMPLISHED BY USING '|~|S' AS THE FIRST CHARAC->  
|cr||nl|  
RUPTED FOR MESSAGES FROM THE CONFERENCE. THIS INTERRUPTION CAN BE|cr->  
||nl|  
PREVENTED BY USING '|~|N' AS THE FIRST CHARACTERS OF INPUT FROM THE|c->  
r||nl|  
TERMINAL. TO RESUME RECEIPT OF MESSAGES FROM THE CONFERENCE THE|cr||->  
nl|  
CHARACTERS '|~|P' WOULD BE USED.|cr||nl||cr||nl|  
 EXAMPLE|cr||nl|  
 COMMAND?PDAC TSS|cr||nl|  
|cr||nl|COMMAND ?~

T12770 |cr||nl|THIS VERB ALLOWS AN INDIVIDUAL TO REQUEST A PRINTED COPY OF S->  
ELECTED|cr||nl|  
PORTIONS OF THE TRANSCRIPT DURING THE CONFERENCE. THE TRANSCRIPT|cr|->  
|nl|  
MAY BE DIRECTED TO A REMOTE PRINTER OR BY DEFAULT BE DIRECTED TO THE|->  
cr||nl|  
SYSTEM PRINTER. SOME VALIDITY CHECKING OF THE ARGUMENTS IS DONE|cr||->  
nl|  
PRIOR TO SPAWNING THE PRINT JOB. IF AN ERROR IS FOUND THE|cr||nl|

PARTICIPANT WILL BE INFORMED AND THE REQUEST IGNORED. THE ARGUMENTS|->  
 cr||nl|  
 ARE:|cr||nl|  
 PRINT ALL -- ALL CONFERENCE MESSAGES WILL BE PRINTED|cr||nl->  
 |  
 PRINT ALL BY PARTICIPANT NAME -- ALL MESSAGES BY THE NAMED|->  
 cr||nl|  
 PARTICIPANT WILL BE PRINTED|cr||nl|  
 PRINT LAST XX -- THE MOST CURRENT XX MESSAGES WILL BE|cr||n->  
 l|  
 PRINTED|cr||nl|  
 PRINT LAST XX BY PARTICIPANT NAME -- THE MOST CURRENT XX|cr->  
 ||nl|  
 MESSAGES BY THE NAMED PARTICIPANT WILL BE PRINTED|cr||->  
 nl|  
 PRINT N[;N...] -- THE SPECIFIED MESSAGES WILL BE PRINTED|cr->  
 ||nl|  
 THE LIST SHOULD NOT EXCEED 20 NUMBERS|cr||nl|  
 PRINT N-X -- MESSAGES 'N' THROUGH 'X' WILL BE PRINTED|cr||->  
 nl|  
 PRINT SUBJECT KEYWORD -- ALL MESSAGES WITH A SUBJECT LINE|c->  
 r||nl|  
 CONTAINING THE SPECIFIED KEYWORD WILL BE PRINTED|cr||n->  
 l|  
 PRINT FLOOR -- ALL FLOOR ACTIVITY WILL BE PRINTED|cr||nl|  
 PRINT FLOOR FROM HHMM TO HHMM ON DDMMYY -- ALL ACTIVITY|cr|->  
 |nl|  
 RECORDED DURING THE TIME SPAN ON THE SPECIFIED DAY|cr|->  
 |nl|  
 WILL BE PRINTED|cr||nl|  
 PRINT FLOOR ON DDMMYY -- ALL FLOOR ACTIVITY RECORDED ON|cr->  
 ||nl|  
 THE SPECIFIED DAY WILL BE PRINTED|cr||nl|  
 PRINT EVENTS -- RECORDED EVENTS WILL BE PRINTED|cr||nl|  
 PRINT MESSAGES -- CONFERENCE MESSAGES WILL BE PRINTED|cr||n->  
 l||cr||nl|  
 TO DIRECT THE OUTPUT TO A REMOTE PRINTER THE PHRASE 'ON ID' WOULD|cr|->  
 |nl|  
 FOLLOW THE REQUEST. 'ID' WOULD BE A VALID RMOTE PRINTER|cr||nl|  
 DESIGNATOR.|cr||nl||cr||nl|  
 EXAMPLE 1|cr||nl|  
 COMMAND?PRINT ALL ON AA|cr||nl||cr||nl|  
 EXAMPLE 2|cr||nl|  
 COMMAND?PRINT FLOOR ON 26OCT80|cr||nl|  
 |cr||nl|COMMAND ?"

T12780 |cr||nl|THIS VERB DISCONNECTS A PARTICIPANT FROM THE CONFERENCE PROGR->  
 AM.|cr||nl||cr||nl|  
 EXAMPLE|cr||nl|  
 COMMAND?QUIT|cr||nl|

|cr||nl|COMMAND ?^

T12790 |cr||nl|THIS VERB ALLOWS A SELECTIVE REVIEW OF CONFERENCE MESSAGES AN->  
D|cr||nl|  
FLOOR ACTIVITY. SELECTIVITY IS PROVIDED BY THE ARGUMENTS THAT|cr||nl->  
|  
ACCOMPANY THE VERB. THESE INCLUDE|cr||nl|  
REVIEW ALL -- ALL PREVIOUS MESSAGES WILL BE REVIEWED.|cr||n->  
l|  
REVIEW ALL BY PARTICIPANT NAME -- ALL MESSAGES BY THE|cr||n->  
l|  
NAMED INDIVIDUAL WILL BE REVIEWED.|cr||nl|  
REVIEW LAST XX -- THE MOST CURRENT XX MESSAGES WILL|cr||nl|  
BE REVIEWED.|cr||nl|  
REVIEW LAST XX BY PARTICIPANT NAME -- THE MOST CURRENT|cr||->  
nl|  
XX MESSAGES BY THE NAMED INDIVIDUAL WILL BE|cr||nl|  
REVIEWED.|cr||nl|  
REVIEW N[,N...] -- THE SPECIFIED MESSAGES WILL BE|cr||nl|  
REVIEWED. THE LIST CANNOT EXCEED 20 VALID MESSAGE|cr||->  
|nl|  
NUMBERS.|cr||nl|  
REVIEW N-X -- MESSAGES 'N' THROUGH 'X' WILL BE DISPLAYED|cr->  
||nl|  
REVIEW SUBJECT -- THE LIST OF AVAILABLE SUBJECTS FOR|cr||nl->  
|  
TALK MESSAGES WILL BE DISPLAYED.|cr||nl|  
REVIEW SUBJECT KEYWORD -- ALL MESSAGES CONTAINING THE|cr||n->  
l|  
SPECIFIED KEYWORD WILL BE LISTED.|cr||nl|  
REVIEW FLOOR -- ALL RECORDED FLOOR ACTIVITY WILL BE|cr||nl|  
REVIEWED.|cr||nl|  
REVIEW FLOOR FROM HHMM TO HHMM -- ALL FLOOR ACTIVITY|cr||nl->  
|  
RECORDED DURING THE GIVEN TIME SPAN WILL BE REVIEWED.|->  
cr||nl|  
|cr||nl|COMMAND ?^

T12800 |cr||nl|THIS VERB PRODUCES A REPORT OF THE CURRENT CONFERENCE STATUS.->  
THE|cr||nl|  
DISPLAY INCLUDES CONFERENCE NAME, START TIME AND DATE, CURRENT|cr||nl->  
|  
CHAIRMAN'S NAME, HOST NAME, WHETHER THE FLOOR IS OCCUPIED, AND A|cr||->  
nl|  
LIST OF ACTIVE PARTICIPANTS AND THEIR MODE.|cr||nl||cr||nl|  
EXAMPLE|cr||nl|  
COMMAND?STATUS|cr||nl|  
|cr||nl|COMMAND ?^

T12810 |cr||nl|THIS VERB TRANSFERS A PARTICIPANT FROM COMMAND MODE TO TALK M->

ODE.|cr||nl|  
 TALK MODE IS THE STATE IN WHICH A PARTICIPANT CAN BUILD MESSAGES|cr||->  
 nl|  
 TO BE DELIVERED TO OTHER CONFERENCE PARTICIPANTS.|cr||nl||cr||nl|  
     EXAMPLE|cr||nl|  
     COMMAND?TALK|cr||nl|  
 |cr||nl|COMMAND ?^

T12820 |cr||nl|THIS VERB'S USAGE IS RESTRICTED TO THE CHAIRMAN AND ALLOWS HI->  
 M TO|cr||nl|  
 TERMINATE A CONFERENCE SESSION. A TERMINATED CONFERENCE CAN NOT BE|c->  
 r||nl|  
 RECONVENED. TERMINATION WILL NOT TAKE PLACE IF THERE ARE ACTIVE|cr||->  
 nl|  
 PARTICIPANTS OTHER THAN THE CHAIRMAN STILL CONNECTED TO THE|cr||nl|  
 CONFERENCE. NO PARTICIPANTS ARE PERMITTED TO JOIN THE CONFERENCE|cr||->  
 |nl|  
 ONCE A TERMINATE REQUEST IS ENTERED.|cr||nl||cr||nl|  
     EXAMPLE|cr||nl|  
     COMMAND?TERMINATE|cr||nl|  
 |cr||nl|COMMAND ?^

T12830 |cr||nl|THE QUESTION MARK IS USED TO OBTAIN AN EXPLANATION OF THE MOD->  
 E|cr||nl|  
 THE PARTICIPANT IS IN. IT WORKS FROM COMMAND AND TALK MODES.|cr||nl||->  
 |cr||nl|  
     EXAMPLE|cr||nl|  
     COMMAND??|cr||nl|  
 |cr||nl|COMMAND ?^

T13000 |cr||nl||ff||cr||nl|TALK MODE ENTERED IN ^

T13010 |cr||nl|>^

T13011 |cr||nl|-^

T13012 |cr||nl|T0?^

T13013 |cr||nl|FROM?^

T13014 |cr||nl|SUBJECT?^

T13015 |cr||nl|CLASS?^

T13016 |cr||nl|INFO?^

T13017 |cr||nl|REF?^

T13050 \$desc^

T13051 \$comm^

T13052 \$dele^

T13053 \$end^

T13054 \$list^

T13055 \$sele^

T13056 \$subj^

T13057 \$stalk^

T13058 ?|cr|^

T13100 |cr||nl||ff||cr||nl|>>>\$DESCRIBE<<<  
|cr||nl|^

->

T13101 |cr||nl||ff||cr||nl|>>>\$COMMAND<<< |cr||nl|^

T13102 |cr||nl||ff||cr||nl|>>>\$DELETE<<< |cr||nl|^

T13103 |cr||nl||ff||cr||nl|>>>\$END<<< |cr||nl|^

T13104 |cr||nl||ff||cr||nl|>>>\$LIST<<< |cr||nl|^

T13105 |cr||nl||ff||cr||nl|>>>\$SELECT<<|cr||nl|^

T13106 |cr||nl||ff||cr||nl|>>>\$SUBJECT<<< |cr||nl|^

T13107 |cr||nl||ff||cr||nl|>>>\$TALK<<< |cr||nl|^

T13108 |cr||nl||ff||cr||nl|>>> ? << |cr||nl|^

T13109 |cr||nl||ff||cr||nl|>>>\$INSERT<<< |cr||nl|^

T13099 |cr||nl|INVALID DESCRIBE ARGUMENT PASSED^

T13110 |cr||nl|THIS PARAMETER IS USED TO OBTAIN AN EXPLANATION OF THE POSSIB->  
LE |cr||nl|  
TALK MODE VERBS. THE AVAILABLE VERBS ARE |cr||nl||cr||nl|  
\$COMMAND \$DELETE \$DESCRIBE \$END \$INSERT |->  
cr||nl|  
\$LIST \$SELECT \$SUBJECT \$TALK ? |cr||->  
nl||cr||nl|  
THE PARAMETER TO BE EXPLAINED WOULD FOLLOW \$DESCRIBE. |cr||nl||cr||->  
nl|

EXAMPLE |cr||nl|  
>\$DESCRIBE \$END |cr||nl|^



T13120 |cr||nl|THIS PARAMETER WILL CAUSE A MESSAGE BEING BUILT TO BE ACCEPTE->  
 D AS |cr||nl|  
 COMPLETE AND QUEUED FOR TRANSMISSION. THE PARTICIPANT WILL THEN BE |->  
 cr||nl|  
 TRANSFERRED TO COMMAND MODE. |cr||nl||cr||nl|^

T13130 |cr||nl|THIS PARAMETER WILL CAUSE THE MESSAGE BEING ENTERED TO BE ->  
 |cr||nl|  
 DELETED AND THE TERMINAL TRANSFERRED TO LISTEN MODE.|cr||nl|^

T13140 |cr||nl|THIS PARAMETER WILL CAUSE A MESSAGE BEING BUILT TO BE ACCEPTE->  
 D |cr||nl|  
 AS COMPLETE AND QUEUED FOR TRANSMISSION. THE MESSAGE WILL BE |cr||->  
 nl|  
 ASSIGNED A NUMBER AND STAMPED WITH THE AUTHOR'S NAME AND THE TIME ->  
 |cr||nl|  
 AND DATE. THE PARTICIPANT WILL THEN BE TRANSFERRED TO LISTEN MODE. |->  
 cr||nl||cr||nl|  
 (NOTE: NULL INPUT(CARRIAGE RETURN) FROM A TERMINAL WILL SERVE THE |->  
 cr||nl|  
 SAME PURPOSE. |cr||nl|^

T13150 |cr||nl|\$LIST AS THE FIRST CHARACTERS OF INPUT WILL CAUSE THE ENTIRE ->  
 MESSAGE|cr||nl|  
 BEING BUILT, OR THE LAST SEVERAL LINES ENTERED IF THE \$LIST IS |->  
 cr||nl|  
 FOLLOWING BY A NUMBER, TO BE LISTED.|cr||nl||cr||nl|  
 EXAMPLE |cr||nl|  
 >\$LIST 10 |cr||nl|^

T13160 |cr||nl|THIS VERB ALLOWS A PARTICIPANT TO DIRECT HIS MESSAGE TO UP TO->  
 FOUR |cr||nl|  
 SPECIFIED RECIPIENTS. IF \$SELECT IS NOT SPECIFIED THE MESSAGE IS |->  
 cr||nl|  
 DELIVERED TO ALL ACTIVE PARTICIPANTS|cr||nl||cr||nl|  
 EXAMPLE |cr||nl|  
 >\$SELECT PARTICIPANT1 PARTICIPANT2 PARTICIPANT3 |cr||nl||->  
 cr||nl|^

T13170 |cr||nl|THIS VERB ALLOWS A PARTICIPANT TO ASSIGN A SUBJECT LINE TO A|->  
 cr||nl|  
 MESSAGE. THE LINE SHOULD CONTAIN A WORD THAT IS IN THE SUBJECT |cr||->  
 nl|  
 LIST CREATED BY THE CONFERENCE CHAIRMAN IF THE MESSAGE IS TO BE |cr||->  
 nl|  
 REVIEWED BY SUBJECT, OTHERWISE NOT REQUIRED.|cr||nl||cr||nl|  
 EXAMPLE |cr||nl|  
 >\$SUBJECT LANTCOM MANPOWER RESOURCES |cr||nl||cr||nl|^

T13180 |cr||nl|THIS PARAMETER WILL CAUSE A MESSAGE BEING BUILT TO BE ACCEPTE->

D AS |cr||nl|  
 COMPLETE AND QUEUED FOR TRANSMISSION. THE PARTICIPANT WILL REMAIN|->  
 cr||nl|  
 IN TALK MODE, AND BE PROMPTED TO START ANOTHER MESSAGE. |cr||nl||cr||->  
 nl|

EXAMPLE |cr||nl|  
 >\$TALK|cr||nl|^

T13190 |cr||nl|THE QUESTION MARK IS USED TO FIND OUT THE CURRENT MODE. BESID->  
 ES |cr||nl|  
 STATING THE MODE, IT GIVES A BRIEF DESCRIPTION OF THE MODE'S |->  
 cr||nl|  
 FUNCTION. -

T13200 |cr||nl|THIS PARAMETER ALLOWS PREVIOUSLY CREATED FILES TO BE INSERTED->  
 INTO A|cr||nl|  
 MESSAGE. THE FILE MUST BE IN STANDARD FORMAT WITH TIMESHARING |cr||->  
 nl|  
 ASCII MEDIA CODE 6. THE LOGICAL RECORD SIZE SHOULD BE 80 CHARACTERS|->  
 cr||nl|  
 OR LESS. THE FULL NAME OF THE FILE TO BE INSERTED MUST FOLLOW THE |->  
 cr||nl|  
 PARAMETER. THE ACTUAL NUMBER OF LINES THAT WILL BE ACCEPTED IS SET |->  
 cr||nl|  
 BY THE CHAIRMAN. THIS PARAMETER MAY ALSO BE USED TO INSERT A |cr||->  
 nl|  
 PREVIOUS MESSAGE INTO THE ONE BEING BUILT. THIS IS DONE BY NOT FOL->  
 cr||nl|  
 LOWING THE REQUEST WITH A FILE NAME. A PROMPT WILL FOLLOW FOR THE |->  
 cr||nl|  
 NUMBER OF THE MESSAGE TO BE INSERTED. |cr||nl||cr||nl||cr||nl|

EXAMPLE |cr||nl|  
 >\$INSERT UMC/CATA/FILE |cr||nl|^

T13210 |cr||nl||ff||cr||nl|THIS TERMINAL IS CURRENTLY IN TALK MODE. ->  
 |cr||nl||cr||nl|  
 THE FUNCTION OF TALK MODE IS THE ENTERING OF MESSAGES FOR OTHER |->  
 cr||nl|  
 PARTICIPANTS ON THE CONFERENCE. A MESSAGE MAY HAVE A SUBJECT |cr||->  
 nl|  
 ASSIGNED TO IT, HAVE A PREEXISTING FILE INCLUDED IN IT, OR HAVE A |->  
 cr||nl|  
 PREVIOUS TALK MESSAGE INCLUDED IN IT. THE TALK MODE FUNCTIONS ARE |->  
 cr||nl|  
 CONTROLLED BY THE FOLLOWING VERBS |cr||nl||cr||nl|  
 \$COMMAND \$DELETE \$DESCRIBE \$END \$INSERT |->  
 cr||nl|  
 \$LIST \$SELECT \$SUBJECT \$TALK ? |cr||->  
 nl||cr||nl|  
 AN EXPLANATION OF EACH PARAMETER IS OBTAINED BY USING THE |c->

r||nl|  
PARAMETER '\$DESCRIBE' FOLLOWED BY THE VERB TO BE EXPLAINED. |cr||nl||->  
cr||nl|

EXAMPLE |cr||nl|  
>\$DESCRIBE \$INSERT

|cr||nl||cr||nl||->

cr||nl|  
THE EXIT FROM TALK UPON COMPLETION OF A MESSAGE IS CONTROLLED BY THE|->  
cr||nl|  
USER. CONTROL IS BASED ON THE TALK MODE VERB USED TO END THE MESSA->  
GE. |cr||nl|~

T13300 |cr||nl|MESSAGE DELETED BY PARTICIPANT ^

T13320 |cr||nl|MESSAGE ACCEPTED^

T13340 |cr||nl|NO TEXT HAS BEEN ENTERED^

T13360 |cr||nl|ILLEGAL SELECT REQUEST DUE TO--NO PARTICIPANT NAMES|cr||nl|^

T13380 |cr||nl|~^

T13381 |cr||nl|INVALID SUBJECT REQUEST--NO SUBJECT LINE |cr||nl|>^

T13420 |cr||nl|MESSAGE NUMBER ^

T13430 |cr||nl||ff||cr||nl|CONTINUING IN TALK MODE ^

## APPENDIX B

### ENVIRONMENT MANAGER PROCESS (EMP) USER INTERFACE

The Environment Manager Process (EMP) allows FSM personnel, the Environment Manager (EM), to monitor and change the FSM operating environment. The EM can observe a user's current status, reassign users to Screeners, and change FSM screen modes and bandwidth thresholds. The EM initiates EMP by logging on as a normal UNIX user. After logging on, the EM is initially presented with a help message informing him of the available commands. The EMP then signifies that it is ready to accept commands by printing a "\*" on the terminal.

EMP provides the EM with seven commands. These commands allow the EM to make all necessary changes in the FSM environment. The commands are:

1. ASSIGNMENTS - Display a list of current logged on users and Screeners and the assignments of each.
2. DISPLAY - Display detailed user/system information.
3. HELP - Display a list of commands or detailed information about a specific command.
4. HISTORY - Invoke the ATDP program to view previous FSM events.

5.    MODIFY - Change the current FSM operating environment.
6.    PROFILE - Display the current status of logged on users.
7.    QUIT - Terminate EMP and log off UNIX.

Descriptions of each command follow, including examples and syntax. Also, the syntax for the complete set of EMP commands is provided in Appendix A.

#### B.1 ASSIGNMENTS

The ASSIGNMENTS command provides the EM with a list of all current user/Screeners assignments. It displays all current users and Screeners logged on, showing which Screener is assigned to which user. If a user currently has no Screener, ASSIGNMENTS displays "No Screener" for that user. If a Screener currently has no users, ASSIGNMENTS displays "- none assigned -" for that Screener. The command syntax is:

a[ssignments] <nl>

where the letters enclosed in brackets "[]" are optional and <nl> represents a newline or carriage return. ASSIGNMENTS returns to the command level prompt ("\*") after displaying the user/Screener assignments.

Following is an example of the output from the ASSIGNMENTS command.

CURRENT FSM USER/SCREENER ASSIGNMENTS

Mon Sep 15 14:56:28 1980

<u>Screener</u>	<u>User</u>
mees	af3gwbrglre
atlas	- none assigned -
mees	af3gwbrgmes
jones	af9gwglmrmb
jones	af3gwbrgslw

B.2 DISPLAY

The DISPLAY command gives the EM a detailed description of current FSM system state as well as the current states of any logged on users. At the system level, DISPLAY produces the current bandwidth rate, the screening status for each WWMCCS subsystem, the number of Screeners currently logged on, and the number of users currently logged on. At the user level, the DISPLAY command gives the user identification, the user's current terminal, the assigned Screener, the current bandwidth rate, and the screening status for each WWMCCS subsystem. The syntax for this command is:

```
d[isplay] [<userlist>] <nl>
<userlist> := <user> | <user> <userlist>
<user>      := system | <username>
<username>  := a valid logged on WWMCCS user name
<nl>        := a newline or carriage return
```

Arguments enclosed in brackets "[ ]" are optional. DISPLAY returns to the command prompt ("\*") after displaying the requested information. A

maximum of five arguments to the DISPLAY command is allowed.

The DISPLAY command with no arguments causes reports on the system and all logged on users to appear. For example:

```
dis <nl>
```

might cause EMP to display:

CURRENT FSM ENVIRONMENT VALUES

Mon Sep 22 09:51:54 1980

System-wide :

Bandwidth is 10 messages/minute

Subsystem screening status:

ACCESS is normal

LIST is normal

SIOS is normal

TCON is normal

TLCF is normal

TSS is normal

WWDMS is normal

Screeners logged on - 1

Users logged on - 1

User:

Username - af3gwbrgmes

Terminal - 7

Screeners assigned - erickson

Bandwidth is 5 messages/minute

Subsystem screening status:

ACCESS is normal

LIST is none

SIOS is normal

TCON is none

TLCF is normal

TSS is all

WWDMS is normal

Had the EM input:

```
d sys <nl>
```

EMP would have displayed only the system portion of the above display. Alternatively, the EM could have asked for information on users. For example, if the EM inputs:

```
di af3gwbrgjdm af8gwglmrmb af3gwglmjtc <nl>
```

EMP would output:

CURRENT FSM ENVIRONMENT VALUES

Mon Sep 22 09:51:54 1980

User:

```
Username - af3gwbrgjdm
Terminal - k
Screener assigned - jones
Bandwidth is 10 messages/minute
Subsystem screening status:
  ACCESS is none
  LIST   is none
  SIOS   is normal
  TCON   is normal
  TLCF   is all
  TSS    is none
  WWDMS  is normal
```

User:

```
Username - af8gwglmrmb
Terminal - 1
Screener assigned - jones
Bandwidth is 1 messages/minute
Subsystem screening status:
  ACCESS is none
  LIST   is none
  SIOS   is none
  TCON   is none
  TLCF   is none
  TSS    is none
  WWDMS  is none
```

User:

```
Username - af3gwglmjtc
Terminal - j
Screener assigned - jones
Bandwidth is 25 messages/minute
Subsystem screening status:
```



```

ACCESS is none
LIST   is all
SIOS   is normal
TCON   is none
TLCF   is none
TSS    is all
WWDMS  is normal

```

Finally, the EM could have asked for users and system information, as in the example:

```
dis af3gwbrgrmb system af9gwglmslw <nl>
```

which would result in a display of the system information, af3gwbrgrmb's information, and finally the information on af9gwglmslw.

If the EM enters an invalid user name (not defined to FSM or not currently logged on) EMP outputs an appropriate message and reprompts for the information.

### B.3 HELP

The HELP command displays information about all or each EMP command. The command syntax is:

```

he[lp] [<command-name-list>] <nl>
<command-name-list>  := <command-name> |
                        <command-name><command-name-list>
<command-name>       := assignments | display | help |
                        history | modify | profile | quit
<nl>                 := newline or return key

```

Syntax is shown in a modified BNF format. Optional arguments are shown in brackets "[ ]", and terminal tokens are shown between the characters "<>".

The bar character "|" separates options.

The HELP command displays general information on all commands or the syntax for one or more specific commands. If the EM entered:

```
h <nl>
```

EMP would display:

This is a brief list of EMP commands. To get further information on a specific command, enter "help <command-name>" where the valid command names are "assignments, display, help, history, modify, profile, quit".

ASSIGNMENTS	- List current user/Screeners assignments.
DISPLAY	- Display current screening and bandwidth data for system and all logged on users or specified data only.
HELP	- Display this list of commands or information about one or more specified commands.
HISTORY	- Call the Audit Trail Display Program to display audit data.
MODIFY	- Modify specified system or user data.
PROFILE	- Display current profile data for all or specified users.
QUIT	- Terminate the Environment Manager's Program.

In another example, if the EM entered:

```
help display modify <nl>
```

EM would output:

DISPLAY	- Display current screening and bandwidth data for system and all logged on users or specified data only.
---------	---

Syntax is shown in a modified BNF format. Optional arguments are shown in brackets "[ ]", and terminal tokens are shown between the characters "<>". The bar character "|" separates options.

Syntax:

```
display [<disp-arg-list>] <nl>
```

<disp-arg-list> := system | <user-name-list> | system <user-name-list>  
 <user-name-list> := <user-name> | <user-name> <user-name-list>  
 <user-name> := <valid user name known to FSM (WES user)>  
 where only five arguments are allowed.

<nl> := newline or return key

MODIFY - Modify specified system or user data.

Syntax is shown in a modified BNF format. Optional arguments are shown in brackets "[ ]", and terminal tokens are shown between the characters "<>". The bar character "|" separates options.

Syntax:

modify [<modify-arg>] <nl>  
 where a menu will be presented with the various choices  
 if no arguments are specified.

<modify-arg> := <mod-screen-arg> | <mod-band-arg> |  
 <mod-sub-arg> | <mod-uscreen-arg> |  
 <mod-uband-arg> | <mod-usub-arg> |  
 <mod-assign-arg>

<mod-screen-arg> := screen <screen-mode>  
 <mod-band-arg> := bandwidth <bandwidth-rate>  
 <mod-sub-arg> := subsystem <subsystem-name> <screen-mode>  
 <mod-uscreen-arg> := user <user-name> <screen-mode>  
 <mod-uband-arg> := ubandwidth <user-name> <bandwidth-rate>  
 <mod-usub-arg> := usubsystem <user-name> <subsystem-name>  
 <screen-mode>  
 <mod-assign-arg> := uassignments <user-name> <Screener-name>

<screen-mode> := all | normal | none | off  
 where none and off mean no screening.

<bandwidth-rate> := digit between the values 1 and 255 inclusively.  
 <subsystem-name> := tss | tlcfr | wwdms | tcon | sios | access | list  
 <user-name> := valid user name known to FSM (WES user)  
 <Screener-name> := valid user name known to FSM (Screener)

<nl> := newline or return key

#### B.4 HISTORY

The HISTORY command allows the EM to view previous FSM events that were audited. Its invocation activates the Audit Trail Display Program (ATDP). ATDP permits the display of any and all audit events. The syntax of the HISTORY command is:

```
hi[story] <nl>
```

where the letters in brackets "[ ]" are optional, and <nl> represents a newline or carriage return. A complete description of the options available for ATDP appears in Section 3. Control returns to EMP when the EM is finished with ATDP.

#### B.5 MODIFY

The MODIFY command is the most powerful of all EMP commands. Through it the EM can modify the FSM environment. Environment values that may be modified include:

1. system-wide screening
2. system-wide bandwidth threshold
3. subsystem screening
4. user screening
5. user subsystem screening

6. user bandwidth threshold

7. user/Screenener assignments

Because of the complexity of this command, two forms are available. The long form accepts arguments and processes one MODIFY command at a time; it can be used by an EM familiar with EMP. The other, the short form, does not accept arguments and prompts the user for the necessary information. The initial prompt is a menu requesting the user to enter the number of the MODIFY command he/she desires. Further prompts are specific questions. This form of the MODIFY command continues until the user specifically exits from it.

#### B.5.1 Short Form Modify Command

The short form MODIFY command has the syntax:

```
m[odify] <nl>
```

where the letters enclosed in brackets "[ ]" are optional and <nl> represents a newline or carriage return. EMP responds with

```
MODIFY CURRENT FSM ENVIRONMENT VALUES      Mon Sep 15 15:36:45 1980
```

Please enter the character which corresponds to the desired action

- 1 - modify system-wide screening
- 2 - modify system-wide bandwidth
- 3 - modify subsystem screening
- 4 - modify user screening
- 5 - modify user/subsystem screening
- 6 - modify user bandwidth
- 7 - modify user/Screenener assignments
- h - display this list of options
- q or <cr> - terminate modifications

action?

The EM enters the character at the "action?" prompt. An incorrect response, "22" for example, causes EMP to output:

"22" is an invalid argument for this command.

**B.5.1.1 Short Form Modify System-Wide Screening.** If the EM wants to change system-wide screening he enters "1<nl>" in response to the "action?" menu prompt. EMP responds with:

MODIFY System-Wide Screening

Mon Sep 15 15:37:21 1980

Set screening to all, none, or normal (q or <cr> to quit):

The EM may enter "all", "off", "none" or "normal" in response to this prompt, setting system-wide screening to all, none or normal respectively. The characters "off" may be used at any time a screen mode is requested and means the same as "none". The EM may also enter any part of the above strings as long as enough characters are provided to uniquely identify it. Entering "nor" for example causes EMP to respond:

Current system-wide screening is now normal

The EM may also hit newline or return in response to the above prompt.

In this case EMP says:

System-wide screening not changed

Had the EM entered an incorrect response, for example "nominal", EMP indicates this fact and prompts for another user input:

"nominal" is an invalid screen mode. Enter replacement or <cr>

If the EM enters an ambiguous screen mode, "no" for example, EMP says:

Input no is ambiguous. Enter replacement or <cr>

**B.5.1.2 Short Form System-Wide Bandwidth.** To change the system bandwidth threshold the EM enters "2<nl>" in response to the menu prompt. EMP will then output:

MODIFY System-wide Bandwidth Tue Sep 16 11:21:45 1980

Current system-wide bandwidth is 10 messages/minute.

Set it to (1-255) (q or <cr> to quit):

The EM may enter any integer within the specified range, "5" for example, and EMP will output:

Current system-wide bandwidth is now 5

An incorrect entry, "350" for example, will cause:

"350" is an invalid bandwidth rate. Enter replacement or <cr>

At this point the EM may enter a replacement value or return. If a return is entered, EMP says:

Bandwidth not changed

and returns to the menu prompt display.

**B.5.1.3 Short Form Modify Subsystem Screening.** Entry of "3" followed by <nl> at the menu prompt causes EMP to output:

MODIFY Subsystem Screening

Tue Sep 16 11:52:36 1980

Subsystem Screening Status:

ACCESS is normal  
LIST is none  
SIOS is all  
TCON is normal  
TLCF is none  
TSS is normal  
WWDMS is normal

Enter the next subsystem name (q or <cr> to quit):

The EM may then enter any subsystem name (with enough characters to uniquely identify it). Entering q or <cr> causes EMP to return to the menu prompt. If the user enters "w" followed by <nl> (he wants to modify the screen mode for WWDMS), EMP responds with:

Set screening to all, none or normal (q or <cr> to quit):

If EM enters "a" followed by <nl> in response to this query, EMP outputs

Screening for WWDMS is now all

EMP then reprompts for another subsystem name.

The entry of an invalid or ambiguous system name or screen mode causes EMP to issue an appropriate message and reprompt for the information.

**B.5.1.4 Short Form Modify User Screening.** A "4" response followed by <nl> will cause EMP to enter the MODIFY user screening subcommand and output

MODIFY User Screening

Tue Sep 16 11:40:29 1980

Enter the next user name whose overall screen mode you wish to modify  
<q or <cr> to quit):

EM then enters a user name, "af8gwbrgjd" for example. EMP then asks for the screen mode:

Set screening to all, none or normal (q or <cr> to quit):



A correct response (enough characters to identify the string as one of "all", "none" or "normal"), "non" for example, causes EMP to display:

Current screening for af8gwbrqjdm is none

EMP then reprompts for another user name. The characters "all" may be input to change the screen modes for all currently logged on users.

The entry of an invalid username (unknown to FSM or not currently logged-on) causes a reprompt. An invalid or ambiguous screen mode causes an appropriate message and a reprompt.

B.5.1.5 Short Form Modify User Subsystem Screening. If the EM enters "5" followed by <nl> in response to the menu prompt, he/she can then change the screen modes of specified users for specified subsystems. EMP prompts with:

MODIFY User/Subsystem Screening

Tue Sep 16 13:18:36 1980

Enter the next user name whose subsystem screen mode you wish to modify  
(q or <cr> to quit):

The EM then enters a valid user name, "af8gwbrqlre" for example. EMP then prompts for subsystem:

Enter the next subsystem name (q or <cr> to quit):

A valid unique subsystem entry (enough characters to identify the string as one of "all", "access", "list", "sios", "tcon", "tlcf", "tss" or "wdms"), "tl" for example, causes EMP to output:

Set screening to all, none or normal (q or <cr> to quit):

A correct response (enough characters to identify the string as one of "all", "none" or "normal"), "a" for example, causes:

Please enter the character which corresponds to the desired action

- 1 - modify system-wide screening
- 2 - modify system-wide bandwidth
- 3 - modify subsystem screening
- 4 - modify user screening
- 5 - modify user/subsystem screening
- 6 - modify user bandwidth
- 7 - modify user/Screeners assignments
- h - display this list of options
- q or <cr> - terminate modifications

action?

During normal operation, the EMP will continue to output data to the screen until it is full, and then request the EM to hit a newline key to get the next screen. The above options will be repeated each time the screen is cleared.

#### B.5.2 Long Form Modify Commands

The following sections describe each long form MODIFY command in detail. In contrast to the short form, long form MODIFY commands do only the action specified. Repeat MODIFY commands are not allowed in the long form.

B.5.2.1 Long Form Modify System-Wide Screening. This subcommand changes the value of the system screen mode. Its effect is to change the screen mode for all subsystems as well as the screen modes for all logged on user's subsystems (and all users who subsequently log on to FSM). The syntax for this command is

```
m[odify] sc[reen] <mode> <nl>
```

```
<mode> := a[ll] | non[e] | nor[mal] | o[ff]
```

<nl> := return or newline key

Arguments enclosed in brackets "[ ]" are optional.

For example,

```
m scr none <nl>
```

would change the value of the screen mode of all subsystems for all logged on users so that no screening would occur. It would result in the message:

Current system-wide screening is now none

Had the EM entered "screen" incorrectly, EMP would output an appropriate message and return to the command level prompt. Had the EM entered an incorrect or ambiguous screen mode, EMP would output an appropriate message and reprompt.

**B.5.2.2 Long Form Modify System-Wide Bandwidth Threshold.** This subcommand changes the value of the system bandwidth threshold. Its effect is to change the threshold value for all users currently logged on to FSM (and all users who subsequently log on). The syntax for this command is:

```
m[odify] b[andwidth] <rate> <nl>
```

<rate> := positive integer between 1 and 255 inclusive.

<nl> := return or newline key

Arguments enclosed in brackets "[ ]" are optional.

If the EM enters

```
mod b 10 <nl>
```

for example, the command would result in changing the current system-wide bandwidth threshold to 10 and would output:

Current system-wide bandwidth is now 10

If the EM had entered an out-of-range bandwidth EMP would output an appropriate message and reprompt for another bandwidth value.

**B.5.2.3 Long Form Modify Subsystem Screening.** This subcommand changes individual subsystem screen modes. Its effect is to change the individual subsystem screen mode for all users currently logged onto FSM (and all users who subsequently log on). The syntax for this command is:

```
m[odify] su[bsystem] <subname> <mode> <nl>
<subname> := ac[cess] | l[ist] | s[ios] | tl[cf] |
           tc[on] | ts[s] | w[wdms] | al[l]
<mode>    := a[ll] | non[e] | nor[mal] | o[ff]
<nl>      := return or newline key
```

Arguments enclosed in brackets "[ ]" are optional.

An example of this command is:

```
m su tc a <nl>
```

This command would set the screen mode for tcon to all. EMP would output:

```
Screening for TCON is now all
```

If the EM had entered an incorrect or ambiguous subsystem name, or an incorrect or ambiguous screen mode, EMP would issue an appropriate message and reprompt.

**B.5.2.4 Long Form Modify User Screening.** This subcommand changes all subsystem screen modes for a single user. The syntax of the command is:

```
m[odify] use[r] <username> <mode> <nl>
<username> := a valid logged on WWMCCS username | a[ll]
<mode>     := a[ll] | non[e] | nor[mal] | o[ff]
```

<nl> := return or newline key

Arguments enclosed in brackets "[ ]" are optional.

An example of this command is:

```
modify user af3gwbrqlre all <nl>
```

This command would change the screen mode of all subsystems for user af3gwbrqlre to all. EMP would output:

Current screening for af3gwbrqlre is all

If the EM had entered an invalid username (not defined to FSM or not currently logged on) or an incorrect or ambiguous screen mode, EMP would output an appropriate message and reprompt for the information.

**B.5.2.5 Long Form Modify User Subsystem Screening.** This subcommand changes individual screen modes for a single user. The syntax of the command is:

```
m[odify] usu[bsystem] <username> <subname> <mode> <nl>
<username> := a valid logged on WWMCCS username | a[ll]
<subname>  := ac[cess] | l[ist] | s[ios] | tl[cf] |
              := tc[on] | ts[s] | w[wdms] | a[ll]
<mode>     := a[ll] | non[e] | nor[mal] | o[ff]
<nl>       := return or newline key
```

Arguments enclosed in brackets "[ ]" are optional.

An example of this command is:

```
mod usub af8gwglmxyz lis non <nl>
```

This command would change af8gwglmxyz's list subsystem screen mode to none. EMP would output:

Current screening for af8gwglmxyz using LIST is none

Had the EM entered an invalid username (not defined to FSM or not currently logged on), an incorrect or ambiguous subsystem name, or an incorrect or ambiguous screen mode, EMP would output an appropriate message and reprompt.

**B.5.2.6 Long Form Modify User Bandwidth.** This subcommand changes a bandwidth threshold value for a single user. The syntax of the command is:

```
m[odify] ub[andwidth] <username> <rate> <nl>
<username> := a valid logged on WWMCCS username | a[ll]
<rate>      := a positive integer between 1 and 255 inclusive
<nl>        := return or newline key
```

Arguments enclosed in brackets "[ ]" are optional.

An example of this command is:

```
m ub af3gwbrgmes 2 <nl>
```

This command would change the bandwidth threshold for the user af3gwbrgmes to 2. EMP would output in response to this command:

```
Current bandwidth for af3gwbrgmes is now 2
```

If the EM enters an invalid username (not defined to FSM or not currently logged on), or an incorrect bandwidth threshold rate, EMP outputs an appropriate message and reprompts for the information.

**B.5.2.7 Long Form Modify User/Screeners Assignments.** This subcommand changes a user's Screener. The syntax of the command is:

```
m[odify] ua[ssignments] <username> <Screenername> <nl>
```

<username> := a valid logged on WWMCCS username

<Screenername> := a valid logged on Screener

<nl> := return or newline key

Arguments enclosed in brackets "[ ]" are optional.

An example of this command is:

```
mod uas af3gwbrglre mees <nl>
```

This command would change af3gwbrglre's current Screener to be Mees. EMP would output a message indicating the change:

User af3gwbrglre is now assigned to Screener Mees

If the EM enters an invalid username or Screener name (not defined to FSM or not currently logged on) EMP issues an appropriate message and reprompts.

## **B.6 PROFILE**

The PROFILE command displays the current status of any or all current FSM low-side users. There are six status types:

1. waiting on user input
2. waiting on system output

3. waiting on Screener response
4. waiting on user input/system output
5. waiting on Screener assignment
6. waiting to logon (connect to WWMCCS).

Waiting on user input means that the user's FSMGP is waiting for the WWMCCS user to enter data so that it may proceed. Waiting on system output means that the WWMCCS user's FSMGP is waiting for output from WWMCCS to proceed. Waiting for Screener response means that the user's FSMGP is waiting for a decision from a Screener before the system output can be sent to the user. Waiting on user input/system output means that the user's FSMGP is waiting for data from either WWMCCS or the user in order to proceed (e.g., LISTEN mode in TLCF). Waiting on Screener assignment means that the user is waiting to be assigned a Screener before he may proceed. Waiting to logon means that the user's FSMGP is waiting to connect to the WWMCCS via the Remote Network Processor Emulator (RNPE).

The command syntax is:

```
p[profile] [<userlist>] <nl>
<userlist> := <user> | <user> <userlist>
<user>      := <termid> | <username>
<termid>    := last character of valid FSM terminal
<username>  := a valid logged on WWMCCS username
<nl>        := a newline or return key
```

Arguments enclosed in brackets "[ ]" are optional. PROFILE returns to the



command prompt ("\*") after displaying the status information.

The PROFILE command with no arguments causes profiles for all currently logged on WWMCCS users to be displayed. Each logged on user causes one record to be displayed with the user name, terminal ID, current subsystem status and DTG (date-time group) of the status to be displayed. For example:

```
pro af3gwbrgjdm <nl>
```

would cause EMP to output:

```
CURRENT FSM USER STATUS          Wed Sep 17 14:53:46 1980

User      Terminal Subsystem  Waiting On  DTG of Status
af3gwbrgjdm 1          WWDMS    user input  Sep 17 14:30:22 1980
```

In another example, if the EM entered:

```
p af3gwbrgmes 5 af8gwglmlre <nl>
```

EMP might display:

```
CURRENT FSM USER STATUS          Wed Sep 17 15:13:07 1980

User      Terminal Subsystem  Waiting On  DTG of Status
af3gwbrgmes 6          TSS      Screener assignment  Sep 17 15:12:03 1980
af8gwbrgjdm 5          TLCF      system output      Sep 17 15:12:43 1980
af8gwbrglre f          TCON      user input          Sep 17 15:05:40 1980
```

If the EM enters an invalid username (not defined to FSM or not currently logged on) or an invalid terminal identifier (not currently in use) EMP outputs an appropriate message and prompts for the information.

#### B.7 QUIT

The QUIT command terminates EMP and logs the user off the system. Its syntax is:

q[uit] <nl>

where the letters in brackets "[ ]" are optional, and <nl> is a newline or carriage return.

AD-A171 332

FORSYTH SECURITY MONITOR (FSM) COMPUTER PROGRAM  
DEVELOPMENT SPECIFICATION (TYPE B5)(U) LOGICON INC SAN  
DIEGO CA FEB 81 N00039-78-C-0323

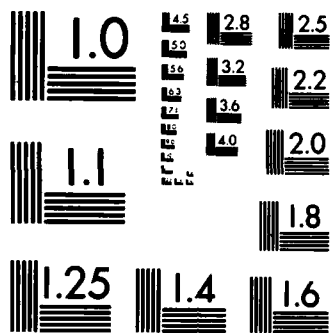
4/4

UNCLASSIFIED

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

## APPENDIX C

### SCREENER TRUSTED PROGRAM (SCTP) USER INTERFACE

The decision to let users view system output is made by Screener personnel via the Screener Trusted Program (SCTP). Since SCTP must be verified, its interface with the Screener is minimal. All inputs are single letter responses to prompts, and SCTP operates in raw mode (no newline or return keystrokes are necessary). Therefore, no syntax is presented in this section. Rather, a scenario is presented in which all possible Screener actions are covered.

The Screener logs on to FSM in the normal UNIX fashion. If there are no downgrades pending, SCTP clears the screen and outputs:

\*\*\*\*\* No Pending Downgrade Requests \*\*\*\*\*

Enter l to logout :

This message remains on the screen until a downgrade is requested or the Screener types "l". When a downgrade request appears, SCTP clears the terminal screen and outputs:

\*\*\*\*\* DOWNGRADE REQUESTED \*\*\*\*\*

Enter <return> to proceed

l to logout :

This prompt is followed by three audible alarms (bells). When the Screener is ready to perform the downgrade he presses the terminal's "return" key. SCTP then outputs the first page of downgrade data followed by the prompt (on the bottom of the same screen):

"downgrade data"

\*\*\*\*\* Downgrade File \*\*\*\*\*

Classification is: OFFICIAL USE ONLY

Enter r for REJECT :

<return> to continue :

The Screener is not allowed to accept a downgrade request until the entire contents of the downgrade file has been displayed; the Screener can always reject a downgrade at any time during his perusal of the data. The above prompt is positioned at the bottom of each page (except the last page) of downgrade information. Notice that the classification of the user's output is displayed for the Screener. This information is obtained from the user's response to the "CLASSIFICATION OF YOUR OUTPUT?" query from WWMCCS. The user in this example responded "ufo". The following is a list of possible classification outputs displayed by SCTP:

1. if the user enters "uzz" SCTP displays "UNCLASSIFIED"
2. if the user enters "ufo" SCTP displays "OFFICIAL USE ONLY"
3. if the user enters "czz" SCTP displays "CONFIDENTIAL"
4. if the user enters "szz" SCTP displays "SECRET"
5. if the user enters "swp" SCTP displays "SECRET WORKING PAPERS"

When the last page of information is displayed SCTP outputs the following prompt:

"downgrade data"

\*\*\*\*\* Downgrade File \*\*\*\*\*

Classification is: OFFICIAL USE ONLY

Enter r for REJECT  
a for ACCEPT  
<return> to continue :

SCTP will clear its Screener's input buffer before the output of each prompt. Thus, the Screener must wait for the data to be presented before replying. In addition, SCTP will not accept invalid responses. Should the Screener enter one, SCTP outputs "???" and waits for the next entry. This is true for all prompts.

For all downgrade requests, the Screener is given the option of viewing an additional context file. This file contains two screens of information just preceding the downgrade data. This file contains both ordered user input and system output. In some situations this additional data may aid the Screener in determining whether to accept or reject the downgrade request. If the Screener decides to view the context file (by depressing the "return" key) then SCTP outputs the first page of data from this file followed by the prompt

"context information data"

\*\*\*\*\* Information File \*\*\*\*\*

Enter r for REJECT  
a for ACCEPT  
<return> to continue :

SCTP always displays downgrade and context data in a cyclic manner. That is, if the Screener always depressed the "return" key, SCTP would first display the downgrade data (one page at a time), and then the context file data (if present), and then cycle back to the downgrade data, and so on.

If the Screener decides to accept a downgrade request (after having viewed all downgrade data) he depresses "a". SCTP requires that the Screener confirm all accept/reject inputs and thus outputs



\*\*\*\*\* Confirm Downgrade \*\*\*\*\*

Enter y            for CONFIRMATION  
         n            for no :

If the Screener enters "n" (for no confirmation), SCTP outputs the next page of the current file (downgrade or context file), followed by the appropriate reject-accept-continue prompt. If the Screener enters "y" (confirming the downgrade request), SCTP outputs

Downgrade Accepted

and proceeds to the next downgrade request. If the Screener decides to reject the downgrade request SCTP asks for confirmation

\*\*\*\*\* Confirm Rejection \*\*\*\*\*

Enter y            for CONFIRMATION  
         n            for no :

If the Screener confirms the rejection SCTP responds with

Downgrade Rejected

and proceeds to the next downgrade request.

When a downgrade request comes as a result of the Environment Manager changing the screen mode to ALL, SCTP displays this fact in the downgrade file prompt:

\*\*\*\*\* Downgrade File (All/EM) \*\*\*\*\*

Classification is: SECRET

Enter r for REJECT

a for ACCEPT

<return> to continue :

If the screen mode had changed because the user exceeded his/her current bandwidth threshold, SCTP displays this in the downgrade file prompt:

\*\*\*\*\* Downgrade File (All/Band) \*\*\*\*\*

Classification is: SECRET WORKING PAPERS

Enter r for REJECT

a for ACCEPT

<return> to continue :

## APPENDIX D

### AUDIT TRAIL DISPLAY PROGRAM (ATDP) USER INTERFACE

The Audit Trail Display Program (ATDP) displays FSM audit events to help analyze FSM activities. The program is invoked by the Environment Manager (EM) via the "history" command. (See Appendix B for more information on EM commands).

The FSM Audit Trail consists of a "current" file and saved files that were previously current. When a current file is saved (by using the END command), its name will be changed from 'current' to the date-time group (DTG) of the save. In this way the size of the current file may be limited to a manageable amount. In addition, if the EM saved the current file daily, a uniform audit data base could be maintained. See the END command for more details.

ATDP commands include:

1. END - Save the current audit file and start a new one.
2. FILES - List all of the existing audit files.
3. HELP - Display a list of commands or information about one or more specified commands.

4. LIST - Display audited events with optional specifiers in ascending time order.

5. QUIT - Terminate the Audit Trail Display Program.

Additionally, unrecognizable input causes a message to be output and a return to the command level prompt.

When ATDP is invoked by the EM, it initially displays a help message informing the EM of the available commands. Then ATDP signifies that it is ready to accept commands by printing a ">" on the terminal. This informs the user that any of the commands listed above can be entered. This prompt distinguishes it from other EM commands, which prompt with an asterisk (\*). The following sections describe each ATDP command in detail.

#### D.1 END

The END command saves the current audit file.

The command syntax is:

e[nd] <nl>

<nl> := newline or return key

The END command terminates the current audit file and saves it. The current file will be renamed with the DTG of the save. This command allows the audit files to be kept to reasonable sizes, allowing more efficient processing by the LIST command. To illustrate the renaming

property of END, assume that the current audit file is ENDED on April 2 at 12:05:30. The current audit file would be renamed to "0402120530" and the "current" file length would be reset to zero.

## D.2 FILES

The FILES command lists all existing audit files.

The command syntax is:

```
files [<output-flag>] <nl>
```

```
<output-flag>    := -t | -l | -p
```

```
-t = output to terminal with paging (default)
```

```
-l = output to terminal with no paging
```

```
-p = output to printer with no paging
```

```
<nl>              := newlinn or return key
```

The FILES command displays in order all past audit files that have been saved by the END command. It displays each file's name, DTG of creation, date of last modification and size in 512 character blocks. In addition, FILES displays the above information on the "current" file. If the FILES command were entered as:

```
f <nl>
```

ATDP would output, for example:

File name	Creation date	Last modified date	Size (blocks)
0904115524	Tue Aug 19 15:15:53 1980	Thu Sep 4 11:03:29 1980	250
0906101828	Thu Sep 4 11:55:26 1980	Sat Sep 6 10:16:09 1980	52
0908083713	Sat Sep 6 10:18:29 1980	Mon Sep 8 05:43:59 1980	60
0908114403	Mon Sep 8 08:37:14 1980	Mon Sep 8 10:37:35 1980	19
0908143618	Mon Sep 8 11:44:04 1980	Mon Sep 8 14:34:10 1980	33
0909084141	Mon Sep 8 14:36:19 1980	Tue Sep 9 08:32:49 1980	1
0909140811	Tue Sep 9 08:42:04 1980	Tue Sep 9 13:57:03 1980	68
0910145636	Tue Sep 9 14:08:13 1980	Wed Sep 10 14:55:02 1980	71
current	Wed Sep 10 14:56:37 1980	Fri Sep 19 13:03:30 1980	22

### D.3 HELP

The HELP command displays information about all or each ATDP command.

The command syntax is:

```
help [<output-flag>] [<command-name-list>] <nl>
```

```
<output-flag>      := -t | -l | -p
```

```
-t = output to terminal with paging (default).
```

```
-l = output to terminal with no paging.
```

```
-p = output to printer with no paging.
```

```
<command-name-list> := <command-name> | <command-name> <command-name-list>
```

```
<command-name>     := end | files | help | list | quit
```

```
<nl>                := newline or return key
```

Syntax is shown in a modified BNF format. Optional arguments are shown in brackets "[ ]", and terminal tokens are shown between the characters "<>".

The bar character "|" separates options.

The HELP command displays general information on all commands or the syntax for one or more specific commands. If the EM entered:

h <nl>

ATDP would display:

This is a brief list of ATDP commands. To get further information on a specific command, enter "help <command-name>" where the valid command names are "end, files, help, list, quit".

END - Save the current audit file and start a new one.  
FILES- List all of the existing audit files.  
HELP - Display this list of commands or information about one or more specified commands.  
LIST - Display audited events with optional specifiers in ascending time order.  
QUIT - Terminate the Audit Trail Display Program.

In another example, if the EM entered:

help files <nl>

ATDP would output:

FILES - List all of the existing audit files.

Syntax is shown in a modified BNF format. Optional arguments are shown in brackets "[ ]", and terminal tokens are shown between the characters "<>". The bar character "|" separates options.

Syntax:

files [<output-flag>] <nl>

<output-flag> := -t | -l | -p  
-t = output to terminal with paging (default).  
-l = output to terminal with no paging.  
-p = output to printer with no paging.

<nl> := newline or return key

#### D.4 LIST

The LIST command prints all or specified FSM audit events.

The command syntax is:

```
list [<output-flag>] [<list-arg-list>] <nl>
```

```
<output-flag>      := -t | -l | -p
```

-t = output to terminal with paging (default).

-l = output to terminal with no paging.

-p = output to printer with no paging.

```
<list-arg-list>    := <list-arg> | <list-arg> <list-arg-list>
```

```
<list-arg>         := <user-arg> | <system-arg> | <event-arg> | <time-arg>
```

where each <list-arg> can only appear once.

```
<user-arg>         := user [<user-name-list>]
```

```
<user-name-list>   := <user-name> | <user-name> <user-name-list>
```

```
<user-name>        := valid user name known to FSM (manager, screener, WES u
```

```
<system-arg>       := subsystem [<subsys-name-list>]
```

```
<subsys-name-list>:= <subsys-name> | <subsys-name> <subsys-name-list>
```

```
<subsys-name>      := tss | tlcf | wwdms | tcon | sios | access | list | log
```

```
<event-arg>        := event [<event-type-list>]
```

```
<event-type-list> := <event-type> | <event-type> <event-type-list>
```

```
<event-type>       := 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
```

```
13 | 14 | 15 | 16 | 17 | 18 | 19
```

1 = valid user input received

2 = invalid user input received



- 3 = system output sent to user
- 4 = system output not sent to user
- 5 = system output accepted by screener
- 6 = system output rejected by screener
- 7 = screener logged onto FSM
- 8 = screener logged off of FSM
- 9 = WES user logged onto FSM
- 10 = WES user logged off of FSM
- 11 = user initiated WWMCCS subsystem
- 12 = user assigned to screener
- 13 = user input simulated
- 14 = screen mode modified
- 15 = bandwidth threshold modified
- 16 = environment manager logged onto FSM
- 17 = environment manager logged off of FSM
- 18 = system output received from WWMCCS
- 19 = system output received by screener

<time-arg> := <start-time> | <end-time> | <start-time> <end-time>

<start-time> := start | <dtg>

<end-time> := end | <dtg>

where a <dtg> by itself will be interpreted as a <start-time>.

<dtg> := MMDDhhmm[YY]

MM = month (01 - 12)

DD = day of month (01 - 31)

hh = hour of day (00 - 23)

mm = minute of hour (00 - 59)

YY = year minus 1900 (00 - 99)

(default time-range is current day)

<nl>               := newline or return key

Syntax is shown in a modified BNF format. Optional arguments are shown in brackets "[ ]", and terminal tokens are shown between the characters "<>". The bar character "|" separates options.

LIST allows the user-oriented display of accumulated FSM audit events. Each event listed contains the following information:

- 1) time of the event
- 2) event description
- 3) event data items such as user name, terminal ID, etc.

When no arguments are entered with the command, LIST produces a listing of all events logged for the current day. Events of particular interest (such as WWMCCS subsystem usage or a specific user history) can also be selectively listed using an optional list argument. See below for sample LIST input and output.

The following arguments are known to LIST:

EVENT event-type(s)	All audit events of the numeric type(s) (see a detailed description of FSM audit events for the event codes and their
---------------------	---

explanation). For instance, all "Valid user input received" events could be listed using an event-type of 1. If EVENT is used with no event-type, the user will be prompted for the desired event-type(s). Input can consist of multiple types.

SUBSYSTEM subsystem(s)      Audit events pertaining to the particular WWMCCS subsystem(s) (i.e. TSS, TLCF, WWDMS, TCON, SIOS, ACCESS, LIST and LOGON) are listed.

USER user-name(s)            Audit events involving the particular user(s) are listed.

The output of LIST may be further restricted to events within a specified time-range. The time-range arguments must be in one of the following forms:

START                        All events from the start of the FSM audit trail to the current time.

END                          All events for the current day.

START                        END                        All events from the start of the FSM audit trail to the current

time.

START            MMDDhhmm[YY] All events from the start of the  
FSM audit trail to the specified  
month, day, hour, minute and year.

MMDDhhmm[YY] MMDDhhmm[YY] All events from the specified  
start-time to the specified  
end-time.

MMDDhhmm[YY] END            All events from the specified month,  
hour, minute and year to the current  
time.

MMDDhhmm[YY]            All events from the specified month,  
day, hour, minute and year to the  
current time.

If arguments are combined, LIST output can become very specific. The  
boolean property of the LIST command allows the EM to restrict the LIST  
output to precise events. For example:

```
list event 11 user af3gwbrgjdm sub sios <nl>
```

would list (for the current day) only the events in which the user  
af3gwbrgjdm entered the WWMCCS subsystem (event 11) SIOS. That is, LIST  
would search for audit events that were event 11 AND user af3gwbrgjdm AND  
subsystem SIOS. Had the ATDP user entered:

```
list event 15 user af3gwbrgjdm af8gwbrgrmb <nl>
```

LIST would search for audit events that were event 15 (bandwidth threshold modified) AND user af3gwbrgjd OR event 15 AND user af8gwbrgrmb. In this way the ATDP user can display each event in which two users had their bandwidth thresholds modified (if any).

Sample LIST input follows:

a) list

This is the default form of LIST; i.e. all events for the current day (April 15) are displayed. This form of the command is equivalent to: list -t 04150000 end

b) list -p event 01

Send all "Valid user input received" (Event 01) audit events for the current day to the line printer.

c) list event start end

Please choose one or more of the following event types:

- 1 - Valid user input received
- 2 - Invalid user input received
- 3 - System output sent to user
- 4 - System output not sent to user
- 5 - System output accepted by screener
- 6 - System output rejected by screener
- 7 - Screener logged onto FSM
- 8 - Screener logged off of FSM
- 9 - WES user logged onto FSM
- 10 - WES user logged off of FSM
- 11 - User initiated WWMCCS subsystem

- 12 - User assigned to screener
- 13 - User input simulated
- 14 - Screen mode modified
- 15 - Bandwidth threshold modified
- 16 - Environment manager logged onto FSM
- 17 - Environment manager logged off of FSM
- 18 - System output received from WWMCCS
- 19 - System output received by screener

Event types? 7 8 9 10

Display on the terminal (with paging) the logging events Screener logon/logoff and User logon/logoff for the entire time-range the audit files cover. Notice, the user was presented with a menu of event-types and prompted for the events desired since none were entered.

d) list -l sub tlcf 04021200 04150830

Display on the terminal (without paging) all audit events concerned with the WWMCCS Teleconferencing (TLCF) subsystem between April 2nd at noon until April 15th at 8:30 am.

e) list user

User name(s)? RAH MEES SCHELLER

Display on the terminal (with paging) all audit events concerned with the named users (RAH, MEES, SCHELLER) for the current day.

f) list -p start end

List all audit events on the printer. Every audited event would be printed out using this command.

Sample LIST output for each audit event is shown in Figure D-1.

```
Wed Sep 10 14:17:23 1980
  Event:      Valid user input received          1
  User:       af8gwbrgjd
  Terminal:   k
  Subsystem:  LOGON
*****      TEXT      *****
001
*****      END OF TEXT      *****
```

```
Wed Sep 10 15:17:19 1980
  Event:      Invalid user input received        2
  User:       af8gwbrgjd
  Terminal:   k
  Subsystem:  TSS
  Reason:     No match.
*****      TEXT      *****
ZZZ
*****      END OF TEXT      *****
```

```
Wed Sep 10 14:17:37 1980
  Event:      System output sent to user        3
  User:       af8gwbrgjd
  Terminal:   k
  Subsystem:  TSS
*****      TEXT      *****
CLASSIFICATION OF YOUR OUTPUT?
*****      END OF TEXT      *****
```

```
Fri Sep 12 12:43:38 1980
  Event:      System output not sent to user    4
  User:       af8gwbrgjd
  Terminal:   1
  Subsystem:  TSS
*****      TEXT      *****
CP DISCONNECT
*****      END OF TEXT      *****
```

```
Wed Sep 10 14:21:18 1980
  Event:      System output accepted by screener 5
  User:       af8gwbrgjd
  Terminal:   k
```

Subsystem: TSS  
 Screener: screener  
 \*\*\*\*\* TEXT \*\*\*\*\*

DATE  
 \*\*\*\*\* END OF TEXT \*\*\*\*\*

Fri Sep 19 11:46:13 1980  
 Event: System output rejected by screener 6  
 User: af3gwbrgrah  
 Terminal: k  
 Subsystem: LIST  
 Screener: harvey  
 \*\*\*\*\* TEXT \*\*\*\*\*  
 ready  
 \*\*\*\*\* END OF TEXT \*\*\*\*\*

Wed Sep 10 14:18:22 1980  
 Event: Screener logged onto FSM 7  
 User: screener  
 Terminal: 7

Wed Sep 17 15:58:26 1980  
 Event: Screener logged off of FSM 8  
 User: harvey  
 Terminal: k

Wed Sep 10 14:17:34 1980  
 Event: WES user logged onto FSM 9  
 User: af8gwbrgjdjdm  
 Terminal: k

Wed Sep 10 08:16:33 1980  
 Event: WES user logged off of FSM 10  
 User: af8gwbrgjdjdm  
 Terminal: l

Wed Sep 10 14:17:36 1980  
 Event: User initiated WWMCCS subsystem 11  
 User: af8gwbrgjdjdm  
 Terminal: k  
 Subsystem: TSS

Wed Sep 10 14:18:26 1980  
 Event: User assigned to screener 12  
 User: af8gwbrgjdjdm  
 Terminal: k  
 Screener: screener  
 Terminal: 7



```

Wed Sep 10 14:17:45 1980
  Event:      User input simulated      13
  User:       af8gwbrgjd
  Terminal:   k
  Subsystem:  TSS
*****      TEXT      *****
zzz
*****      END OF TEXT      *****

Fri Sep 19 12:31:10 1980
  Event:      Screen mode modified      14
  User:       af3gwbrgrah
  Subsystem:  ALL
  Screen mode: No screening

Fri Sep 19 12:25:10 1980
  Event:      Bandwidth threshold modified 15
  User:       ALL
  Bandwidth threshold: 1

Wed Sep 10 14:22:03 1980
  Event:      Environment manager logged onto FSM 16
  User:       manager
  Terminal:   1

Fri Sep 19 13:03:05 1980
  Event:      Environment manager logged off of FSM 17
  User:       manager
  Terminal:   l

Wed Sep 10 14:17:29 1980
  Event:      System output received from WWMCCS 18
  User:       af8gwbrgjd
  Terminal:   k
  Subsystem:  LOGON
*****      TEXT      *****
TERMINAL
*****      END OF TEXT      *****

Wed Sep 10 14:19:22 1980
  Event:      System output received by screener 19
  User:       af8gwbrgjd
  Subsystem:  TSS
  Screener:   screener
  Terminal:   7

```

Figure D-1. Sample LIST output.

Notice several features of Figure D-1. First, each event has an English description of the event as well as the event number. Second, each event has the DTG of its recording. Finally, where applicable, the audit texts are included.

It is also possible to restrict the ATDP output by requesting a combination of arguments. For example, if a user wanted certain events for a given user he might input:

```
l user af3gwbrgjdM event 10 <nl>
```

which would display all events for the current day of type 10 for user af3gwbrgjdM. ATDP might output:

```
List events between Tue Sep 23 08:00:00 1980 and Tue Sep 23 15:19:57 1980
Concerning users: af3gwbrgjdM
Concerning events: 10
```

```
Tue Sep 23 08:16:33 1980
Event:      WES user logged off of FSM          10
User:       af3gwbrgjdM
Terminal:   l
```

Notice that ATDP uses a Boolean algorithm to obtain the user's request. That is, ATDP searches the audit files for events that involve user 'af8gwbrgjdM' AND are of type 10.

In another example, a user might want to restrict the event selection further:

l user af8gwbrgjd event 3 sub sios <nl>

ATDP might output:

List events between Tue Sep 23 08:00:00 1980 and Tue Sep 23 15:38:28 1980  
Concerning users: af8gwbrgjd  
Concerning subsystems: sios  
Concerning events: 3

Tue Sep 23 14:21:36 1980

Event:	System output sent to user	3
User:	af8gwbrgjd	
Terminal:	k	
Subsystem:	SIOS	

*****	TEXT	*****
SIOS ROUTINE		
*****	END OF TEXT	*****

Finally, ATDP outputs an appropriate message when it cannot find the requested events. For example, if no events had been generated for September 23, and the EM entered:

l <nl>

ATDP would output:

List events between Tue Sep 23 00:00:00 1980 and Tue Sep 23 15:47:31 1980 with no specific search categories specified.

No audit records were found which satisfied the requirements.

0.5 QUIT

The QUIT command terminates ATDP processing.

The command syntax is:

q[uit] <nl>

<nl> := newline or return key

QUIT terminates ATDP and returns control to the Environment Manager Process.

## APPENDIX E

### GLOBAL LOGON FILE MODIFIER (USER) USER INTERFACE

The USER utility program adds and deletes users from the FSM environment. With it, WWMCCS users, Screeners and Environment Managers can be added and deleted, and a current list of users (all those listed above) can be displayed. The USER program is executed by issuing the command "user <nl>" at the command prompt. USER responds with the prompt:

```
Please choose the next action to be performed:
Type a to add a new WWMCCS user
Type s to add a new Screener
Type e to add a new Environment Manager
Type d to delete a WWMCCS user
Type t to delete a Screener or Environment Manager
Type l to list currently defined users
Type q to quit the UPDATE program
```

Option?

If the user wants to add a WWMCCS user, he/she enters "a" at the Option? prompt. USER then outputs:

```
Add a WWMCCS user
Please enter a user name with exactly 11 characters with the format:

af[389] [gx] [mw] [site code (3 chars)] [user initials (3 chars)]:
```

If the user enters "af3gwbrgrmb", USER responds with:

```
User af3gwbrgrmb is now a valid user name
```

USER then redisplayes the command options and the prompt. If the user had entered a WWMCCS user name in the wrong format, USER outputs:

```
Invalid user name, must follow the following format:
```

af[389] [gx] [mw] [site code (3 chars)] [user initials (3 chars)]  
and then redisplay the command options prompt. If the user had entered  
too many characters, USER responds with:

Invalid user name, too many characters

and then redisplay the command options prompt. If the user attempted to  
add an existing WWMCCS user (already defined to FSM), 'af3gwbrgjdm' for  
example, USER outputs:

User af3gwbrgjdm is already a valid user name

If the WWMCCS user had previously been deleted (undefined to FSM), USER  
outputs:

User af3gwbrgrmb is a previously used user name.  
Do you wish this name to be reinstated?

A 'y' to this question results in:

User af3gwbrgrmb is now a valid user name

Any other response results in a redisplay of the command options prompt.

If the user wants to add a Screener, he/she would respond to the Option?  
prompt with 's'. USER would respond with:

Add a Screener  
Please enter a user name not exceeding 8 characters.

Had the user entered 'mees', for example, USER would respond with:

User mees is now a valid user name

Had the user entered a name already defined to FSM, or a name with too  
many characters, USER would output an appropriate message and redisplay  
the command options prompt.

If the user wanted to add an Environment Manager to FSM, he/she would enter "e" at the Option? prompt. This would result in:

Add an Environment Manager  
Please enter a user name not exceeding 8 characters:

Had the user entered 'smyth', for example, USER would output:

User smyth is now a valid user name

However, only one Environment Manager may be defined to FSM at any one time. If there had already been an EM when the user tried to add one, USER would output:

Only one manager may be defined at a time, 'smyth' already exists

Deleting (undefining users to FSM) is as easy as adding them. To delete a WWMCCS user the user enters 'd'; to delete a Screener or Environment Manager the user enters 't'. This results in:

Delete a WWMCCS user  
Please enter a user name with exactly 11 characters with the format:

af [389] [gx] [mw] [site code (3 chars)] [user initials (3 chars)]:

for WWMCCS users and

Delete a Screener or Environment Manager  
Please enter a user name not exceeding 8 characters:

for Screeners and EMs. Entering the correct names (they must exist to FSM), results in USER confirming the deletion. Entry errors (invalid format or trying to delete non-existent users) cause USER to issue an error and redisplay the command options prompt.

The user can also display the currently existing FSM users. This is done by entering 'l' at the Option? prompt. USER might display, for example:

USER ID	USER NAME	USER TYPE	VALID USER?	LOGGED ON	TTY ID	ASSIGNED SCREENER
1	smyth	Env Mngr	Yes	No		
2	mees	Screeener	Yes	Yes	1	
3	af3gmbrglre	WES User	Yes	Yes	2	mees
4	af8xwbrgabc	WES User	Yes	No		
5	af8gwbrgjzm	WES User	Yes	No		
6	af9xwglmrmb	WES User	Yes	Yes	5	jones
7	af3gwbrgrah	WES User	Yes	No		
8	jones	Screeener	Yes	Yes	6	
9	af8gwbrgjdm	WES User	Yes	Yes	7	mees
10	af3gwbrgrmb	WES User	No			

Finally, to terminate the USER program, the user enters "q" at the

Option? prompt. USER responds with:

UPDATE program terminated  
Good-bye



## APPENDIX F

### FSMGTTY PROGRAM DESCRIPTION

FSMGTTY is a small program that sets up the WES user/Guard environment when the WES user initiates a terminal session. During FSM startup, FSMGTTY is executed and attempts to open each terminal that is connected to Guard (via the PDP 11/70). After setting terminal attributes to resemble normal WWMCCS terminals, FSMGTTY waits until the WES user enters data at the terminal keyboard. Once this occurs, FSMGTTY fork/executes a copy of itself which obtains current copies of the context tables and fork/executes the Guardian Process (FSMGP). In this way, little machine overhead is consumed while a terminal is not being used, due to FSMGTTY's small size.

FSMGTTY, when executed, changes the UNIX terminal attributes to resemble WWMCCS terminals. These changes involve changing the character delete and line erase characters, as well as mapping keyboard carriage returns into strict carriage returns (sans line feed). It also arranges to ignore potentially troublesome input, such as control-S, delete and other characters that have a special meaning to the UNIX operating system.

When a WES initiates a terminal session (by typing a carriage return or control-A from a tty terminal or "\$\*\$log24" from a vip terminal), FSMGTTY locks the context files, preventing any changes from being made, and

makes copies of them for this user. This facilitates context table changes during normal FSM operation and assures that each WES user obtains the most current version of the context tables. When the copies have been made, the lock is removed and the Guardian Process is fork/executed. Following WES user logoff (either normally, or following a system error), FSMGP terminates and control returns to FSMGTTY. It will start the entire sequence again, beginning with changing the terminal attributes.

An external file (/fsm/gttys) contains terminal identifiers and all associated information that FSMGTTY needs. A copy of this file follows:

```
: FSM active terminals for low-side WES users. Format is:
: one character terminal type, 'v' = vip, 't' = tty/wang.
: Colon to separate fields, ':'.
: one character to be used for logging for this terminal.
: Colon to separate fields, ':'.
: UNIX pathname for terminal, E.g. /dev/tty8.
: Colon to separate fields, ':'.
: UNIX pathname for RNPE line, E.g. /dev/rnpe/300/8
:
t:e:/dev/ttye:/dev/rnpe/300/e
t:f:/dev/ttyf:/dev/rnpe/300/f
t:g:/dev/ttyg:/dev/rnpe/300/g
t:h:/dev/ttyh:/dev/rnpe/300/h
```

END

10286

DTIC